

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی شاهرود

دانشکده مهندسی برق و رباتیک

گروه الکترونیک

فشرده‌سازی سیگنال صوت مبتنی بر فن انطباق الگو

امین قنبرزاده

اساتید راهنما:

دکتر سید علی سلیمانی ایوری

دکتر هادی گرایلو

استاد مشاور:

دکتر حسین مروی

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

بهمن ۱۳۹۱

ب



شماره : ۱۰۴۸/آ.ت.ب

تاریخ : ۹۱/۱۱/۲۸

ویرایش : -----

بسمه تعالی

فرم صورتجلسه دفاع پایان نامه تحصیلی دوره کارشناسی ارشد

با تأییدات خداوند متعال و با استعانت از حضرت ولی عصر (عج) جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای :
امین قنبر زاده رشته : برق
تحت عنوان : فشرده سازی سیگنال صوت مبتنی بر فن انطباق الگو
که در تاریخ ۱۳۹۱/۱۱/۲۸ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح زیر است :

قبول (با درجه : بسیار - امتیاز : ۱۸,۲۵) دفاع مجدد مردود

۱- عالی (۱۹ - ۲۰) ۲- بسیار خوب (۱۸/۹۹ - ۱۸)

۳- خوب (۱۶ - ۱۷/۹۹) ۴- قابل قبول (۱۴ - ۱۵/۹۹)

۵- نمره کمتر از ۱۴ غیر قابل قبول

عضو هیأت داوران	نام و نام خانوادگی	مرتبه علمی	امضاء
۱- استاد راهنما	سید علی میرزا علیزاده	استاد	
۲- استاد مشاور	---	---	---
۳- نماینده شورای تحصیلات تکمیلی	علیرضا احمدی فرد	استاد	
۴- استاد ممتحن	رضا باهری	استاد	
۵- استاد ممتحن	ایده مریدی	استاد	

رئیس دانشکده :

تیمی خاکسارانه به:

پدر و مادرم

تعهد نامه

اینجانب امین قسریزاده دانشجوی دوره کارشناسی ارشد رشته الکترونیک - سیستم دانشکده مهندسی برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان نامه با عنوان :

فشرده سازی سیگنال صوت مبتنی بر فن انطباقی

تحت راهنمایی آقای دکتر سید امین و دکتر گرابلو متعهد می شوم :

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است .
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است .
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است .
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه، در مواردی که از موجود زنده (یا بافت های آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ : ۹۱/۱۱/۲۹

امضاء دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

* متن این صفحه نیز باید در ابتدای نسخه های تکثیر شده پایان نامه وجود داشته باشد .

چکیده

با توجه به محدود بودن پهنای باند در ارتباطات مخابراتی و اینترنت، و محدودیت حافظه‌های در دسترس، و از طرفی نیاز روز افزون به داده‌های صوتی و تصویری با کیفیت بالا، و همچنین به منظور استفاده‌ی بهینه از امکانات موجود، نیازمندی ما به روش‌های فشرده‌سازی سیگنال، اجتناب‌ناپذیر است.

وجلت‌ها یکی از ابزارهای موفق برای فشرده‌سازی و حذف نویز در تصاویر هستند. وجلت‌ها مشابه موجک‌ها یک تحلیل چندمقیاسی از سیگنال را فراهم می‌کنند. در این پایان‌نامه، تعمیم روش فشرده‌سازی مبتنی بر وجلت‌های توسعه یافته، روی سیگنال‌های صوتی، مورد مطالعه قرار گرفته است. برای این کار ابتدا مروری بر روش فشرده‌سازی وجلت‌های اولیه و وجلت‌های توسعه یافته، انجام داده و مشکلات تعمیم این روش را بر روی سیگنال‌های صوتی بررسی کرده، و راه‌کارهای ممکن را به کار گرفته‌ایم. توابع پایه‌ی مربوط به وجلت، توابع چندجمله‌ای از مراتب مختلف انتخاب شده‌اند. با توجه به این که ضرایب به دست آمده، مربوط به حوزه‌ی زمان می‌باشد، و همچنین به دلیل همبستگی موجود بین این ضرایب، سه روش مختلف کوانتیزاسیون را آزمایش کرده‌ایم. یک روش فشرده‌سازی مبتنی بر نواحی مطلوب نیز در ترکیب با روش پیشنهادی، ارائه شده است که باعث می‌شود در سیگنال‌های گفتار، قسمت‌های صحبت و سکوت، با نرخ بیت و کیفیت متفاوتی فشرده شوند. همچنین عملکرد سیستم در محیط‌های نویزی نیز بررسی شده است. از قابلیت‌های روش پیشنهادی این است که نرخ فشرده‌سازی و همچنین کیفیت سیگنال بازسازی شده، قابل تنظیم می‌باشد.

برای ارزیابی، روش پیشنهادی با دو روش مبتنی بر تبدیل DWT و DCT مورد مقایسه قرار گرفته است. نتایج از لحاظ معیارهای کمی SNR و MOS مقایسه شده‌اند. نتایج نشان می‌دهد که از دید معیار SNR روش مبتنی بر وجلت در مورد سیگنال‌های صحبت به طور متوسط ۱۹/۴۰٪ نسبت به روش DWT و ۱۱/۹۰٪ نسبت به روش DCT، بهتر بوده و در سیگنال‌های صوتی نیز به طور

متوسط ۱۹/۹۷٪ نسبت به روش DWT عملکرد بهتری داشته است. نتایج حاکی از این است که روش مبتنی بر وجلت می‌تواند روش مؤثری برای فشرده‌سازی انواع مختلف سیگنال‌های صوتی و صحبت باشد.

کلمات کلیدی: فشرده‌سازی صوت، وجلت، کدگذاری، کدگشایی، فشرده‌سازی مبتنی بر نواحی مطلوب، انطباق الگو.

فهرست مطالب

فصل ۱- مقدمه	۱
۱-۱- مقدمه	۲
۲-۱- صوت دیجیتال و نرخ بیت	۳
۳-۱- کاربردهای فشرده‌سازی صوت	۴
۴-۱- اهداف پایان نامه	۵
۵-۱- ساختار پایان نامه	۶
فصل ۲- مروری بر روش‌های موجود در فشرده‌سازی	۸
۱-۲- معرفی بعضی ابزارهای معمول در فشرده‌سازی	۹
۱-۱-۲- کوانتیزاسیون	۹
۱-۱-۱-۲- کوانتیزاسیون یکنواخت	۱۱
۲-۱-۱-۲- کوانتیزاسیون غیر یکنواخت	۱۲
۳-۱-۱-۲- کوانتیزاسیون به روش DPCM	۱۳
۲-۱-۲- کدگذاری آنروپی	۱۵
۱-۲-۱-۲- کدگذار حسابی	۱۸
۲-۲-۱-۲- روش رمزگشایی در کدگذار حسابی	۲۰
۳-۲-۱-۲- کدگذاری به روش دوره‌ی تداوم	۲۱
۲-۲- مروری بر روش‌های فشرده‌سازی با اتلاف	۲۲
۱-۲-۲- روش کدینگ سینوسی	۲۲
۱-۱-۲-۲- سیستم بازسازی	۲۵
۲-۱-۲-۲- نتایج روش تجزیه و ساخت سینوسی	۲۸
۲-۲-۲- روش انطباق الگو	۳۰
۱-۲-۲-۲- بازسازی سیگنال کد شده	۳۵
۳-۲- مروری بر روش‌های بدون اتلاف	۳۶
۱-۳-۲- روش Audio Pak	۳۶
۱-۱-۳-۲- پنجره بندی	۳۶
۲-۱-۳-۲- حذف همبستگی	۳۷
۳-۱-۳-۲- کدگذاری آنروپی	۳۸
۲-۳-۲- روش LTAC	۳۹
۳-۳-۲- روش مبتنی بر کدگذاری دوره‌ی تداوم	۴۲
فصل ۳- فشرده‌سازی مبتنی بر تبدیل	۴۵
۱-۳- معرفی تبدیل موجک	۴۶

۴۷	تبدیل موجک	۱-۱-۳
۴۹	تبدیل موجک گسسته یا DWT	۲-۱-۳
۵۴	DCT	۲-۳
۵۵	روش فشرده‌سازی مبتنی بر تبدیل	۳-۳
۵۷	فرآیند کلی طرح فشرده‌سازی مبتنی بر تبدیل	۱-۳-۳
۵۸	روش فشرده‌سازی مبتنی بر تبدیل DWT	۴-۳
۵۹	انتخاب تابع موجک بهینه	3-4-1-
۵۹	تعداد سطوح تجزیه	۲-۴-۳
۶۰	آستانه‌گذاری برای صفر کردن ضرایب	۳-۴-۳
۶۱	کوانتیزاسیون	۴-۴-۳
۶۱	نحوه‌ی کدگذاری نمادهای تکراری	۵-۴-۳
۶۲	کد گذاری آنتروپی	۶-۴-۳
۶۳	روش فشرده‌سازی مبتنی بر تبدیل DCT	۵-۳
۶۵	فصل ۴ - وجلت‌ها	
۶۶	وجلت	۱-۴
۶۹	تجزیه‌ی وجلت‌ها	۱-۱-۴
۷۱	دیکشنری وجلت	۲-۱-۴
۷۳	الگوریتم تجزیه‌ی وجلت (درخت تجزیه‌ی بهینه)	۳-۱-۴
۷۴	وجلت‌های توسعه یافته	۲-۴
۷۴	نمایش دیکشنری وجلت با پارامترهای θ و r	۱-۲-۴
۷۵	دیکشنری وجلت‌های توسعه یافته	۲-۲-۴
۷۷	فصل ۵ - روش پیشنهادی	
۷۸	چند جمله‌ای‌ها	۱-۵
۷۸	تقریب تابع	۱-۱-۵
۷۹	تقریب مینیمم مربع خطای گسسته	۲-۱-۵
۸۱	روش فشرده‌سازی مبتنی بر وجلت	۲-۵
۸۱	فریم بندی	۱-۲-۵
۸۱	درخت تجزیه	۲-۲-۵
۸۳	نحوه‌ی تقریب در داخل پنجره‌ها	۳-۲-۵
۸۴	موازنه - تابع هزینه	۴-۲-۵
۸۵	الگوریتم پیدا کردن درخت تجزیه‌ی بهینه	۵-۲-۵
۸۷	تعیین معیار اتلاف (D)	۶-۲-۵
۸۸	تعیین معیار نرخ بیت (R)	۷-۲-۵
۸۹	کدگذاری ضرایب چندجمله‌ای	۸-۲-۵

۹۱	نحوه‌ی انتخاب نقاط تابع	۹-۲-۵
۹۱	محدود کردن مراتب چندجمله‌ای ها	۱۰-۲-۵
۹۲	کوانتیزاسیون پارامترها	۱۱-۲-۵
۹۳	کدگذاری آنتروپی	۱۲-۲-۵
۹۴	روند نما	۱۳-۲-۵
۹۶	قابلیت حذف نویز	۱۴-۲-۵
۹۷	فشرده‌سازی مبتنی بر نواحی مطلوب	۱۵-۲-۵
۱۰۱	فصل ۶- نتایج	
۱۰۲	پایگاه داده	۱-۶
۱۰۲	معیارهای ارزیابی	۲-۶
۱۰۴	تعیین مقدار J	۳-۶
۱۰۶	نوع سیگنال	۴-۶
۱۰۸	ضریب موازنه	۵-۶
۱۰۹	کوانتیزاسیون	۶-۶
۱۱۱	مقایسه	۷-۶
۱۱۴	بررسی تأثیر نویز	۸-۶
۱۱۷	ارتباط روش پیشنهادی و موجک ها	۹-۶
۱۱۹	فصل ۷- نتیجه‌گیری و پیشنهاد	
۱۲۰	نتیجه‌گیری	۱-۷
۱۲۱	پیشنهاد برای ادامه‌ی کار	۲-۷
۱۲۳	پایگاه داده	
۱۲۴	مراجع	

فهرست اشکال

- شکل ۱-۲. کوانتیزاسیون یکنواخت با $N = 7$ و $\Delta = 1$ ۱۱
- شکل ۲-۲. عملکرد یک چندی‌ساز PCM غیریکنواخت با توابع متراکم کننده و بسط دهنده ۱۲
- شکل ۳-۲. تابع نگاشت در چندی‌ساز غیر یکنواخت ۱۲
- شکل ۴-۲. نمودار بلوکی کوانتیزاسیون پیش بین حلقه-بسته (DPCM) ۱۴
- شکل ۵-۲. مثال از الگوریتم کدگذاری حسابی ۲۰
- شکل ۶-۲. یک مثال از شناسایی پیک‌های تبدیل STFT از روی طیف دامنه ۲۳
- شکل ۷-۲. یک مثال از اتصال داخلی فرکانس‌ها و فرآیند تولد و مرگ ۲۴
- شکل ۸-۲. چند نمونه از توابع چندجمله‌ای برای درون‌یابی فاز [۱۷] ۲۷
- شکل ۹-۲. بلوک دیاگرام سیستم تجزیه و ساخت سینوسی [۱۷] ۲۸
- شکل ۱۰-۲. سیگنال بازسازی شده از روش سینوسی [۱۷]. الف) سیگنال اصلی. ب) سیگنال بازسازی شده ۲۹
- شکل ۱۱-۲. بلوک دیاگرام روش پیشنهادی در [۲۴] ۳۳
- شکل ۱۲-۲. فشرده‌سازی بدون اتلاف، به عنوان ترکیبی از فشرده‌سازی با اتلاف و ارسال سیگنال خطا ۳۹
- شکل ۱۳-۲. بلوک دیاگرام سیستم فشرده‌سازی بدون اتلاف مبتنی بر تبدیل [۲۸] ۴۰
- شکل ۱۴-۲. بلوک دیاگرام فشرده‌ساز بدون اتلاف مبتنی بر تبدیل (LTAC) ۴۱
- شکل ۱-۳. نقاط مربوط به DWT در صفحه‌ی زمان-مقیاس ۵۲
- شکل ۲-۳. الف) فیلتربانک مربوط به تجزیه‌ی DWT. ب) فیلتر بانک مربوط به بازسازی ۵۳
- شکل ۳-۳. نمودار بلوکی الگوریتم کلی فشرده‌سازهای مبتنی بر تبدیل ۵۷
- شکل ۱-۴. یک تصویر باینری با لبه‌هایی که تصویر را به دو بخش سیاه و سفید تقسیم کرده است ۶۷
- شکل ۲-۴. الف) تقریب با یک خط. ب) تقریب با تکه‌های خطی ۶۷
- شکل ۳-۴. یک مثال از یک وجلت ۶۸
- شکل ۴-۴. تمام حالات پیکربندی ممکن از یک رأس واقع بر محیط یک مربع در یک مقیاس خاص ۶۹
- شکل ۵-۴. یک مرحله انشعاب در تجزیه‌ی وجلت ۷۰
- شکل ۶-۴. الف) شکل یک تجزیه‌ی وجلت. هر کدام از وجلت‌ها می‌توانند هم تراز یا غیر هم تراز باشند. ب) درخت چهارگانه مربوط به تجزیه‌ی قسمت الف ۷۱
- شکل ۷-۴. افراز تصویر در سطوح مختلف و رأس‌ها و بعضی لبه‌ها ۷۲
- شکل ۸-۴. نوع دیگر پارامتر گذاری وجلت‌ها ۷۵
- شکل ۹-۴. پارامتر گذاری وجلت‌های توسعه یافته. ۷۵

- شکل ۵-۱. بخشی از یک درخت تجزیه‌ی کامل. در این شکل، هر کدام از گره‌های سطر بالایی به طول $2J$ ، و هر کدام از گره‌های سطر پایینی به طول $2j_{min}$ می‌باشند ۸۳
- شکل ۵-۲. یک مثال از درخت تجزیه‌ی بهینه را نشان می‌دهد. در این جا، بعد از اتمام الگوریتم، گره‌های n_1, n_2, n_3 و n_4 ، که معادل قسمت‌های x_1, x_2, x_3 و x_4 از سیگنال هستند، به عنوان گره‌های بهینه انتخاب شده‌اند. ۸۷
- شکل ۵-۳. روند نمای کلی روش پیشنهادی ۹۵
- شکل ۵-۴. سیگنال سینوسی نویزی و تقریب آن به وسیله‌ی تابع چندجمله‌ای ۹۶
- شکل ۶-۱. بررسی تأثیر مقدار J بر روی سیگنال به نویز و نرخ بیت ۱۰۵
- شکل ۶-۲. نتایج تغییر SNR با تغییر λ ، برای سیگنال‌های مختلف ۱۰۸
- شکل ۶-۳. نتایج تغییر نرخ بیت با تغییر λ ، برای سیگنال‌های مختلف ۱۰۸
- شکل ۶-۴. نحوه‌ی تغییر نرخ بیت و SNR در روش مبتنی بر وجلت (در چهار λ ی مختلف) ۱۰۹
- شکل ۶-۵. مقایسه‌ی نتایج سه روش کوانتیزاسیون ۱۱۰
- شکل ۶-۶. نتایج مقایسه‌ی SNR روی سیگنال‌های صحبت ۱۱۱
- شکل ۶-۷. نتایج مقایسه‌ی SNR روی سیگنال‌های صوتی ۱۱۲
- شکل ۶-۸. نتایج مقایسه‌ی MOS روی سیگنال‌های صحبت ۱۱۳
- شکل ۶-۹. نتایج مقایسه‌ی MOS روی سیگنال‌های صوتی ۱۱۳
- شکل ۶-۱۰. یک مثال از سیگنال‌های بازسازی شده توسط هر سه روش وجلت، DWT و DCT ۱۱۴
- شکل ۶-۱۱. یک مثال از سیگنال نویزی شده با انحراف معیار نویز $\sigma = 0.01$ ۱۱۵
- شکل ۶-۱۲. هیستوگرام طول گره‌های بهینه ۱۱۷
- شکل ۶-۱۳. هیستوگرام مربوط به مراتب چندجمله‌ای‌های بهینه ۱۱۷

فهرست جداول

- جدول ۱-۲. نرخ فشرده‌سازی برای کانال چپ فایل‌های صوتی از [۲۷] در طول فریم‌های مختلف... ۳۷
- جدول ۲-۲. تعداد دفعاتی که هر کدام از سیگنال‌های خطا، انتخاب شده‌اند. ۳۸
- جدول ۳-۲. نتایج فشرده‌سازی روش LTAC در مقایسه با روش بدون اتلاف Shorten ۴۱
- جدول ۴-۲. مقایسه نرخ فشرده‌سازی روش پیشنهادی در [۲] با پنج روش دیگر. ۴۴
- جدول ۱-۴. نتایج فشرده‌سازی تصاویر کارتونی با دو روش تجزیه‌ی وجلت‌ها (WD) و وجلت‌های توسعه یافته (EWD) [۵۸] ۷۶
- جدول ۱-۶. امتیازها در MOS ۱۰۴
- جدول ۲-۶. نتایج بر روی سیگنال‌های مختلف پایگاه داده ۱۰۷
- جدول ۳-۶. مقایسه‌ی نتایج SNR و MOS حاصل از سیگنال نویزی با انحراف معیار نویز $\sigma = 0.005$ ۱۱۵
- جدول ۴-۶. مقایسه‌ی نتایج SNR و MOS حاصل از سیگنال نویزی با انحراف معیار نویز $\sigma = 0.01$ ۱۱۶

فصل اول

مقدمه

۱-۱ - مقدمه

امروزه در این عصر، صوت دیجیتال به سرعت تبدیل به یکی از بخشهای عمده‌ی زندگی روزمره‌ی ما می‌شود. عیب اصلی سیگنال‌های صوتی دیجیتال، نرخ بیت بیش از اندازه‌ای است که برای این سیگنال‌های یک بعدی وجود دارد. این نرخ بیت بالا، ذخیره و انتقال سیگنال‌های صوتی دیجیتال را به شدت غیر مؤثر و کم بازده می‌کند. مزیت اصلی فشرده‌سازی صوت این است که یک راه حل ساده و مؤثر برای کاهش هزینه‌ی ذخیره و انتقال داده، به ازای کاهش کیفیت سیگنال، فراهم می‌کند. با اتلاف ناچیز در کیفیت سیگنال می‌توان ذخیره‌سازی‌های زیادی انجام داد. این برای یک کاربر، بدین معنی است که می‌تواند موسیقی زیادتری را در یک دیسک ذخیره کند و یا می‌تواند صوت با کیفیت بالاتری در طول اتصال مودم خود، دریافت و یا ارسال کند. اگرچه وسایل ذخیره سازی با گنجایش بالا (مانند DVD^۱) و شبکه‌های داده‌ی با نرخ بیت بالا، قابل دسترسی هستند، با وجود این فشرده‌سازی صوت به دلایل مهم و معقولی، مطلوب و ضروری می‌باشد. نرخ فشرده‌سازی‌های بالا برای صوت، باعث می‌شود تا بتوانیم داده‌های بیشتری را (مانند ویدئو) به همراه سیگنال صوتی انتقال داده و یا ذخیره کنیم. در نتیجه تعداد زیادی از کانال‌های صوتی می‌توانند یک جا ارسال شوند، و بنابراین فشرده‌سازی، استفاده‌ی بهتر از منابع و وسایل در دسترس را ممکن می‌سازد. فشرده‌سازی هم‌چنین امکان دسترسی به ابزار و ویژگی‌های بیشتری همچون مقیاس‌پذیری رشته‌ی بی‌تی^۲ را نیز فراهم می‌کند. مقیاس‌پذیری رشته‌ی بی‌تی، هنگامی که سیگنال‌های صوتی را به صورت بلادرنگ^۳ با اینترنت تبادل می‌کنیم یک خاصیت مطلوب و مفید به شمار می‌آید.

یک سیگنال صوتی مقیاس‌پذیر، این امکان را به کاربر می‌دهد که بتواند یک سیگنال صوتی را بسته به پهنای باند اینترنت در دسترس خود، با سطوح کیفی مختلفی دریافت کند. این به این معنی

^۱ Digital Versatile Disc

^۲ Bitstream Scalability

^۳ Real-time

است که کاربرانی که اینترنت کم سرعت دارند هم می‌توانند به صورت بلادرنگ به سیگنال‌های صوتی گوش کنند اما با کیفیت پایین‌تر.

۱-۲- صوت دیجیتال و نرخ بیت

در دو دهه‌ی اخیر، صوت دیجیتال نسبت به آنالوگ، محبوبیت بیشتری پیدا کرده است. یک دلیل این امر، مقاومت بیشتر آن در برابر تخریب و نویز ناشی از ذخیره‌سازی و انتقال است. به عنوان مثال، صوت دیجیتال در مقابل اثرات تداخل امواج در کانال‌های انتقال، امن‌تر بوده و عاری از مشکلاتی مانند نوسانات سرعت که در سیستم‌های ذخیره‌ی نواری داشتیم، می‌باشد [۱]. استاندارد امروزی صوت دیجیتال، کیفیت در حد CD^۱ است. وقتی که به یک CD گوش می‌کنیم، صداهای آزار دهنده‌ی خش خش و شکستگی که از نوارهای مغناطیسی حاصل می‌شوند، وجود ندارند. هم‌چنین گوش کردن به یک قطعه خاص از یک آلبوم بسیار راحت‌تر است.

با توجه به این مزایای CD، وقتی که این محصول در دهه ۱۹۸۰ توسط سونی^۲ و فیلیپس^۳ معرفی شد، رسانه‌های صوتی دیجیتال، سریعاً جایگزین رسانه‌های صوتی آنالوگ شدند. با وجود این، مقدار بسیار بزرگی از داده‌های فشرده نشده که برای صوت دیجیتال لازم بودند، منجر به بار عظیم ذخیره و انتقال شدند. پیشرفت روش‌های فشرده‌سازی صوت و استانداردهای منتج، این بار را تا حد زیادی کاهش داده‌اند. برای محاسبه‌ی نرخ بیت در CD، در نظر می‌گیریم که در کیفیت CD، سیگنال آنالوگ ابتدا با یک مبدل آنالوگ به دیجیتال، با فرکانس ۴۴۱۰۰ Hz نمونه‌برداری می‌شود. هر کدام از این نمونه‌ها، با دقت ۱۶ بیت به صورت یکنواخت کوانتیزه می‌شوند.

بنابراین نرخ بیت در هر کانال صوتی به این شکل به دست می‌آید که به ازای هر ثانیه ۴۴۱۰۰ نمونه داریم که هر نمونه ۱۶ بیت فضا اشغال می‌کند. بنابراین نرخ بیت ۷۰۵۶۰۰ بیت در

^۱ CD Quality

^۲ Sony

^۳ Philips

ثانیه خواهد بود. این نرخ بیت برای اکثر کاربردهای عملی، بیش از اندازه می‌باشد. توجه داریم که بیشترین سرعت مودم‌های تلفنی استاندارد، ۵۶ کیلو بیت بر ثانیه است. برای غلبه بر این موانع عمده، بحث فشرده‌سازی صوت پدیدار شده است.

۱-۳- کاربردهای فشرده‌سازی صوت

کاربردها و منافع زیادی برای فشرده‌سازی سیگنال‌های صوتی وجود دارد، و اجتناب ناپذیر است که این تعداد، افزایش نیز خواهد یافت. بعضی از این کاربردها عبارتند از:

- **ارتباطات تلفن^۱:** استفاده از نرخ فشرده‌سازی صوت بالا در شبکه‌های دیجیتال سرویس مجتمع^۲ (ISDN)، امکان انتقال سیگنال‌های صوتی استریو^۳ را با یک خط ISDN (در سرعت ۱۲۸kbps) فراهم می‌کند. بنابراین این خط می‌تواند به طور همزمان برای انتقال سیگنال‌های ویدئویی برای تلکنفرانس‌ها^۴ مورد استفاده قرار گیرد.
- **ارتباطات رادیویی^۵:** فشرده‌سازی، باعث کاهش توان خروجی فرستنده و کاهش ملزومات پهنای باند، هم برای سیستم‌های مخابراتی زمینی و هم سیستم ماهواره‌ای می‌شود. با این مزایا، انتشاردهنده‌های صوت دیجیتال^۶ که صوت با پهنای وسیع را ارائه می‌کنند، می‌توانند با رادیو FM جایگزین شوند.
- **ذخیره سازی:** طبیعتاً، نرخ بیت‌های پایین، ملزومات حافظه‌ی کم را برای ذخیره سازی داده ایجاد می‌کنند.

¹ Telecommunication

² Integrated Services Digital Network (ISDN)

³ Stereo

⁴ Teleconference

⁵ Radio Links

⁶ Digital Audio Broadcasting (DAB)

- **تلویزیون HDTV^۱:** تکنولوژی فشرده‌سازی، امکان انتقال صوت با کیفیت بالا و همچنین انتقال تعداد زیادی از کانال‌های صوتی را برای تلویزیون فراهم می‌کند. بنابراین HDTV می‌تواند صدای فراگیر مشابه سینما را در خانه پخش کند.
- **اینترنت:** همان طور که گفته شد، برای انتشار سیگنال‌های صوتی در اینترنت، نیازمند فشرده‌سازی هستیم.
- **محصولات کاربری:** فشرده‌سازی صوت منجر به تولید محصولات متنوعی از قبیل مینی دیسک‌های سونی و کاست فشرده‌ی دیجیتال فیلیپ^۲ شده است. محبوبیت فرمت MPEG1-Layer3 منجر به تولید پخش کننده‌های MP۳^۳ قابل حمل و قابل استفاده در خودرو شده است.
- **نهان نگاری:** در [۲] پیشنهاد شده که در هر فریم، بعد از فشرده‌سازی، می‌توان از فضای آزاد شده، برای وارد کردن اطلاعات امنیتی استفاده کرد. به این مورد نهان نگاری^۴ گفته می‌شود.
- **کاهش هزینه:** در مجموع، استفاده‌ی بهینه از حافظه‌های موجود در دسترس، و توانایی ذخیره سازی بیشتر اطلاعات، و همچنین استفاده بهینه از خطوط انتقال و موارد دیگر که بیان شد، برای شرکت‌های مختلف و نیز کاربران، کاهش هزینه را به دنبال دارد.

۱-۴- اهداف پایان نامه

در این پایان نامه یک روش فشرده‌سازی صوتی از نوع فشرده‌سازهای شکل موج^۵ ارائه شده است. ایده‌ی روش پیشنهادی در اصل از بحث وجلت^۶، که یک الگوریتم موفق در پردازش تصویر می‌باشد گرفته شده است. برای تعمیم این بحث به سیگنال‌های صوتی تدابیری اندیشیده شده است تا

^۱ High Definition Television (HDTV)

^۲ Digital Compact Cassette (DCC)

^۳ MP3 Player

^۴ Watermarking

^۵ Waveform Coder

^۶ Wedgelets

سازگار با سیگنال‌های صوتی باشد. روش پیشنهادی هم برای سیگنال‌های صوت و هم صحبت کاربرد دارد. همچنین دو مورد از قدرتمندترین و معمول‌ترین تبدیلات مورد استفاده در فشرده‌سازی سیگنال‌های صوتی (یعنی DWT^1 و DCT^2) مورد بحث و بررسی قرار گرفته‌اند. از این دو روش مبتنی مبتنی بر تبدیل، برای ارزیابی روش پیشنهادی استفاده شده است.

۱-۵- ساختار پایان نامه

بعد از مقدمه، در فصل دوم، ابتدا ملزومات فشرده‌سازی، از جمله روش‌های مختلف کوانتیزاسیون^۳ به همراه روش‌های کدگذاری آنتروپی و بعضی از پارامترهای موردنیاز معرفی شده‌اند. سپس به معرفی و بررسی بعضی روش‌های موجود از جمله روش کدینگ سینوسی و روش انطباق الگو، از انواع فشرده‌سازهای با اتلاف، و همچنین سه مورد از روش‌های فشرده‌سازی بدون اتلاف پرداخته شده است.

در فصل سوم، بعد از معرفی تئوری تبدیل‌های DWT و DCT ، روش فشرده‌سازی مبتنی بر تبدیل با تأکید بر تبدیل‌های DWT و DCT ، به طور کامل تشریح می‌شود. این دو روش برای مقایسه‌ی روش پیشنهادی نیز مورد استفاده قرار گرفته‌اند.

در فصل چهارم، تئوری بحث و جلتهای اصلی برای تصویر آورده شده است.

در فصل پنجم، روش پیشنهادی بیان شده و مورد بحث و بررسی قرار گرفته است.

در فصل ششم، تأثیر پارامترهای مختلف روش پیشنهادی، بحث و بررسی شده و همچنین نتایج پیاده سازی روش پیشنهادی، هم برای سیگنال‌های صوت و هم صحبت با نتایج حاصل از دو روش مبتنی بر تبدیل DWT و DCT مقایسه شده و مورد بحث قرار گرفته‌اند.

¹ Discrete Wavelet Transform

² Discrete Cosine Transform

³ Quantization

در فصل هفتم، نتیجه‌گیری کلی از روش ارائه شده انجام شده و پیشنهاداتی برای ادامه‌ی کار مطرح شده است.

فصل دوم

مروری بر روش‌های موجود در فشرده‌سازی

در این فصل ابتدا یک معرفی پایه از بعضی از معمول‌ترین ابزارهایی که در اکثر روش‌های فشرده‌سازی وجود دارند ارائه می‌دهیم تا یک آشنایی اولیه حاصل شود. سپس مروری بر برخی روش‌های با اتلاف و بدون اتلاف موجود خواهیم داشت.

۲-۱- معرفی بعضی ابزارهای معمول در فشرده‌سازی

۲-۱-۱- کوانتیزاسیون

جنبه‌های تئوری روش‌های کوانتیزاسیون شکل موج^۱، حدود پنجاه سال پیش بیان شدند [۳]. اگر چه ما با سیگنال‌های گسسته در زمان کار می‌کنیم، اما مقادیر نمونه‌ها اعداد حقیقی هستند که به صورت پیوسته در دامنه‌ی سیگنال توزیع شده‌اند. برای ذخیره‌سازی و انتقال داده و همچنین برای رسیدن به هدف فشرده‌سازی، ما نیاز داریم تا این اعداد حقیقی را به یک مجموعه‌ی گسسته، یا نمادهای گسسته نگاشت کنیم. فرآیند نگاشت یک خط حقیقی به یک مجموعه‌ی نمادهای قابل شمارش را، کوانتیزاسیون یا چندی‌سازی می‌گویند.

کوانتیزاسیون باعث می‌شود تا اعداد را با تعداد بیت‌های کمتر ذخیره کرده و همچنین باعث کاهش آنروپی^۲ می‌شود لذا اگر بخواهیم بعد از کوانتیزاسیون از کدگذاری آنروپی هم استفاده کنیم باید موقع کوانتیزاسیون علاوه از نرخ بیت و کیفیت سیگنال، آنروپی سیگنال کوانتیزه شده را نیز در نظر گرفت.

کوانتیزاسیون، می‌تواند به صورت بدون حافظه^۳ یا با حافظه^۴ باشد. در کوانتیزاسیون بدون حافظه، هر کدام از نمونه‌های ورودی به صورت مستقل به یکی از نمادهای خروجی چندی‌ساز، نگاشت می‌شوند و نیازی به اطلاع از مقادیر قبلی و بعدی سیگنال نیست. اما در کوانتیزاسیون با حافظه،

¹ Waveform Quantization

² Entropy

³ Memoryless

⁴ With Memory

نمونه‌ی فعلی براساس مقادیر نمونه‌های قبلی و یا بعدی، کوانتیزه می‌شود. یک مثال از مؤثرترین روش‌های کوانتیزاسیون با حافظه، روش DPCM^۱ است که در این پایان نامه هم مورد استفاده قرار گرفته است.

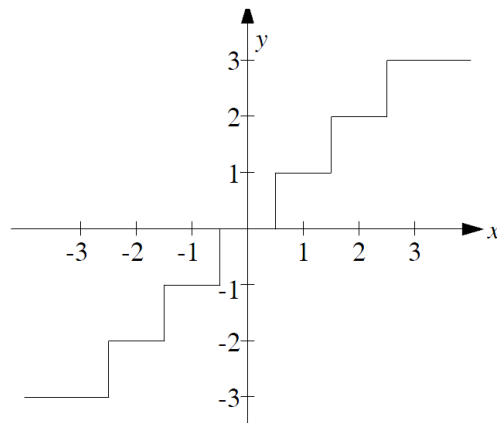
کوانتیزاسیون می‌تواند به صورت یکنواخت^۲ و یا غیریکنواخت^۳ باشد. چندی‌سازهای غیریکنواخت از طول بازه‌های غیریکنواخت برای کوانتیزه کردن استفاده می‌کنند. یک روش معروف و معمول از چندی‌سازهای غیریکنواخت، روش قانون $\mu - law$ می‌باشد که در این پایان نامه از این روش هم استفاده شده است.

کوانتیزاسیون همچنین می‌تواند به دو شکل عددی و برداری می‌باشد. کوانتیزاسیون برداری^۴ زمانی مفید است که بین داده‌ها، همبستگی^۵ وجود داشته باشد. در زیر انواع روش‌های کوانتیزاسیون عددی را توضیح می‌دهیم.

در کوانتیزاسیون عددی^۶، هر کدام از نمونه‌های سیگنال به صورت مستقل، کوانتیزه می‌شود. یک مثال از کوانتیزاسیون عددی، در شکل ۱-۲ نشان داده شده است. محدوده‌ی دامنه‌ی ورودی، به بازه‌های $I_i = (x_{i-1}, x_i]$ افراز شده و مقدار خروجی y_i را به این بازه اختصاص می‌دهیم. به هر کدام از y_i ها نماد می‌گوییم. حال دو بازه نامحدود خواهیم داشت، یعنی بازه‌های $x < \frac{-5}{2}$ و $x > \frac{5}{2}$. اگر تعداد بازه‌ها N باشد، نماد خروجی خواهیم داشت. در این صورت به تعداد $R = \lceil \log_2 N \rceil$ بیت برای نمایش هر y_i لازم خواهد بود.

کوانتیزاسیون یک مرحله‌ی با اتلاف است. بنابراین منظور از عکس عمل کوانتیزاسیون، دقیقاً معکوس مرحله‌ی کوانتیزاسیون نمی‌باشد بلکه منظور از آن انتخاب یک مقدار از بازه‌ی $(x_{i-1}, x_i]$ در صورت ورود نماد y_i می‌باشد.

^۱ Differential Pulse Code Modulation
^۲ Uniform
^۳ Nonuniform
^۴ Vector Quantization
^۵ Correlation
^۶ Scalar Quantization



شکل ۱-۲. کوانتیزاسیون یکنواخت با $N = 7$ و $\Delta = 1$

کوانتیزاسیون عددی می‌تواند به صورت یکنواخت و یا غیر یکنواخت باشد.

۱-۱-۱-۲ کوانتیزاسیون یکنواخت

اگر طول بازه های I_i با هم برابر باشند، کوانتیزاسیون از نوع یکنواخت خواهد بود. هر چقدر طول این بازه‌ها را کوچک‌تر انتخاب کنیم خطای کوانتیزاسیون کمتر خواهد بود. طول این بازه‌ها به سه پارامتر بستگی دارد. [۴].

۱- ماکزیمم مقدار در سیگنال ورودی (M_{max})

۲- مینیمم مقدار سیگنال ورودی (M_{min})

۳- تعداد سطوح کوانتیزاسیون (L)

بعد از این که این پارامترها مشخص شدند، طول هر بازه از رابطه‌ی زیر بدست می‌آید:

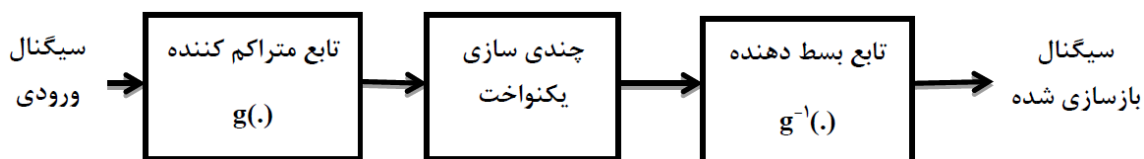
$$\Delta = \left(\frac{M_{max} - M_{min}}{L} \right) \quad (1-2)$$

بنابراین با داشتن این سه پارامتر می‌توان جدول کوانتیزاسیون را به دست آورد. اگر بخواهیم هر نماد را با B بیت نمایش دهیم باید $L = 2^B$ قرار دهیم. در این صورت نمادهای ما می‌توانند اعداد ۰ تا $L - 1$ باشند. این سه پارامتر نیز باید همراه دیگر داده‌ها در فایل ذخیره شوند تا بتوان عکس عمل کوانتیزاسیون را انجام داد [۵].

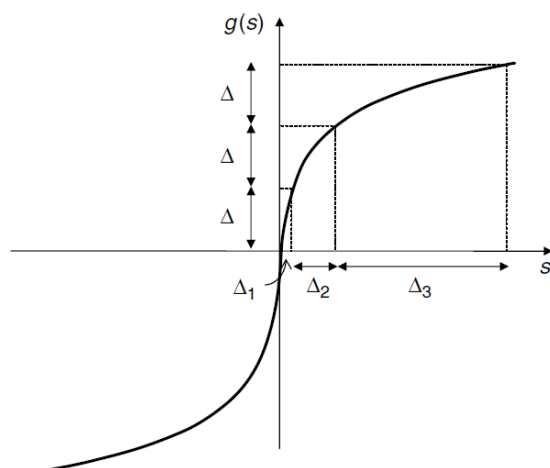
۲-۱-۱-۲- کوانتیزاسیون غیر یکنواخت

کوانتیزاسیون یکنواخت از لحاظ مینیمم مربع خطا برای سیگنال‌هایی با تابع چگالی احتمال یکنواخت بهینه است [۶] اما اگر تابع چگالی احتمال یکنواخت نباشد، کوانتیزاسیون یکنواخت بهینه نیست. چندی‌سازهای غیر یکنواخت از طول بازه‌های غیر یکسان استفاده می‌کنند که این طول بازه‌ها می‌توانند از روی ساختار آماری سیگنال تعیین شوند. چندی‌سازهای غیر یکنواخت برای ورودی‌هایی با مشخصات آماری خاص مناسب هستند و باعث رسیدن به سیگنال به نویز بالا در یک نرخ بیت و تابع چگالی احتمال معلوم می‌شوند [۷].

چندی‌سازهای غیر یکنواخت متنوعی وجود دارد. یکی از مهمترین انواع آن‌ها، چندی‌سازهای لگاریتمی هستند که در کاربردهای تلفن، بسیار معمول می‌باشند [۸، ۹]. نحوه عملکرد یک چندی‌ساز غیر یکنواخت در شکل ۲-۲ به خوبی نشان داده شده است.



شکل ۲-۲. عملکرد یک چندی‌ساز PCM غیر یکنواخت با توابع متراکم کننده و بسط دهنده



شکل ۲-۳. تابع نگاشت در چندی‌ساز غیر یکنواخت

ابتدا یک تابع نگاشت غیر خطی، $g(\cdot)$ ، طول بازه‌های غیر یکنواخت را به طول بازه‌های یکسان نگاشت می‌کند. به طوری که یک چندی‌ساز خطی ساده می‌تواند آن را کوانتیزه کند. یک مثال از تابع نگاشت در شکل ۲-۳ مشاهده می‌شود. سپس هنگام کدگشایی، با استفاده از یک تابع وارون $g^{-1}(\cdot)$ می‌توان سیگنال را بازسازی کرد. دو استاندارد بر اساس چندی‌سازهای لگاریتمی، به نام‌های $\mu - law$ و $A - law$ معرفی شده‌اند. تفاوت زیادی در عملکرد این دو روش وجود ندارد. از روش $\mu - law$ در استاندارد PCM آمریکای شمالی استفاده می‌شود ($\mu = 255$). تابع نگاشت در روش $\mu - law$ به صورت رابطه‌ی (۲-۲) است.

$$y = g(x) = \frac{V \cdot \text{Ln}(1 + \mu|x|/V)}{\text{Ln}(1 + \mu)} \cdot \text{sign}(x) \quad (2-2)$$

که در آن μ ، پارامتر مربوط به این روش و V ، بزرگترین دامنه‌ی بردار x است. وقتی که $\mu = 255$ است، رابطه‌ی (۲-۲)، برای دامنه‌های کوچک، نگاشت تقریباً خطی و برای دامنه‌های بزرگ، نگاشت لگاریتمی دارد. در کاربردهای تلفن، در کیفیت برابر، روش فوق، نرخ بیت را تا اندازه‌ی ۴ بیت بر نمونه، نسبت به چندی‌ساز یکنواخت (PCM) کاهش می‌دهد. به تابع نگاشت $g(\cdot)$ ، متراکم کننده و به $g^{-1}(\cdot)$ بسط دهنده گفته می‌شود. تابع بسط دهنده در روش $\mu - law$ که وارون تابع متراکم کننده است به صورت رابطه‌ی (۳-۲) می‌باشد.

$$x = \frac{V}{\mu} (e^{|\gamma| \cdot \text{Ln}(1+\mu)/V} - 1) \cdot \text{sign}(y) \quad (3-2)$$

۳-۱-۱-۲ کوانتیزاسیون به روش DPCM

یک روش مهم و مفید برای کوانتیزاسیون، زمانی است که به جای کوانتیزه کردن نمونه‌های سیگنال، مقدار اختلاف هر نمونه از سیگنال را، با یک مقدار پیش بینی شده برای آن نمونه، کوانتیزه می‌کنند [۹]. یک روش مؤثر از این نوع کوانتیزاسیون، DPCM است، که تزیاید^۱ موجود در شکل موج

^۱ Redundancy

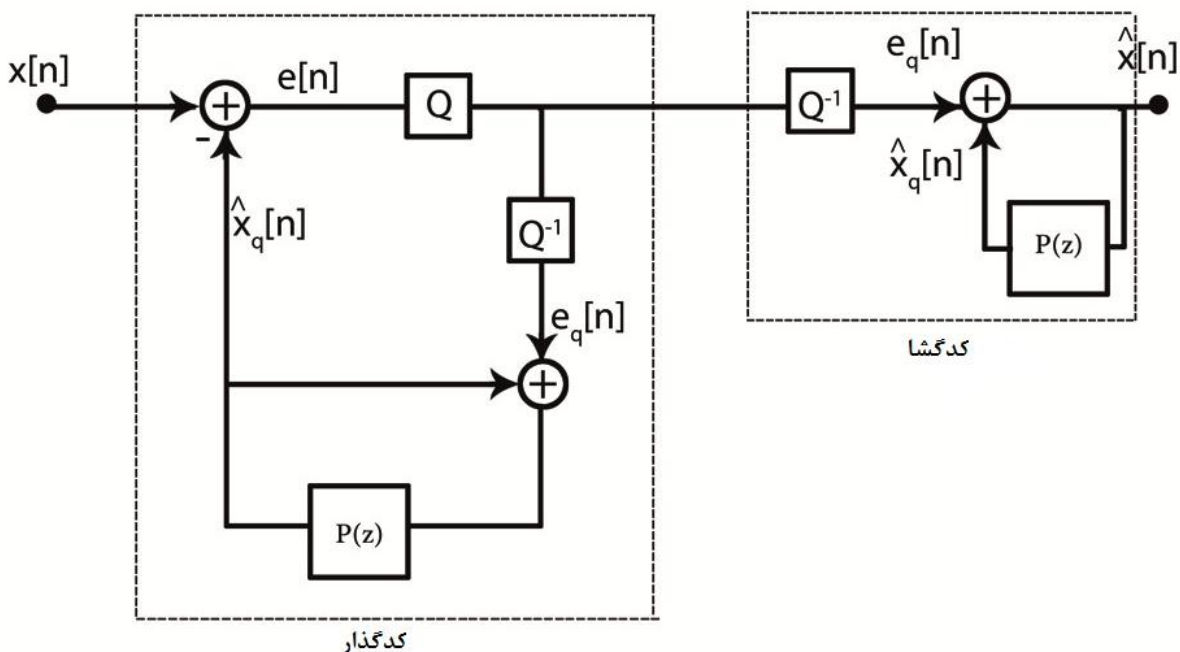
سیگنال‌های صوتی را با بهره‌گیری از همبستگی موجود در بین نمونه‌های متوالی حذف می‌کند [۶]. این روش، جزء روش‌های با حافظه محسوب می‌شود.

در این روش، کدگذار با استفاده از نمونه‌های قبلی سیگنال، نمونه‌ی فعلی را پیش‌بینی می‌کند و سپس اختلاف بین نمونه‌های فعلی و مقدار پیش‌بینی شده ارسال می‌شود. پیش‌بینی‌کننده، یک تابع است که DPCM از آن برای پیش‌بینی نمونه‌ها استفاده می‌کند. یک پیش‌بین خطی به صورت رابطه‌ی (۴-۲) است:

$$y(k) = p(1)x(k-1) + p(2)x(k-2) + \dots + p(m)x(k-m) \quad (4-2)$$

که در آن x سیگنال اصلی و $y(k)$ مقدار پیش‌بینی شده برای $x(k)$ می‌باشد و $p(k)$ ضرایب پیش‌بینی هستند. در حالی که $m = 1$ باشد و $p(1) = 1$ ، در این صورت مقدار پیش‌بینی شده برای

نمونه‌ی فعلی، مقدار نمونه‌ی قبلی خواهد بود. به این حالت خاص DM^۱ گفته می‌شود.



شکل ۴-۲. نمودار بلوکی کوانتیزاسیون پیش‌بین بین حلقه-بسته (DPCM)

¹ Delta Modulation

یک سیستم DPCM همان طور که در شکل ۲-۴ نشان داده شده است به صورت زیر کار می‌کند: مقدار $\hat{x}_q[n]$ مقدار تخمینی برای نمونه‌ی فعلی است و با استفاده از نمونه‌های قبلی سیگنال محاسبه شده است. خطای پیش بینی، $e[n]$ ، کوانتیزه شده و ارسال می‌شود (یعنی $e_q[n]$). خطای پیش بینی کوانتیزه شده دوباره در قسمت کدگذاری به $\hat{x}_q[n]$ اضافه می‌شود تا مقدار $\hat{x}[n]$ را بسازند. در این صورت مقدار پیش بینی شده در قسمت کدگذار (یعنی $\hat{x}_q[n]$)، دقیقاً با مقدار پیش بینی شده در قسمت کدگشا (یعنی $\hat{x}_q[n]$) برابر می‌شوند و بنابراین خطای ایجاد شده در سیگنال فقط مربوط به خطای کوانتیزاسیون سیگنال اختلاف می‌شود. به عبارت دیگر، چون

$$e[n] = x[n] - \hat{x}_q[n]$$

و

$$\hat{x}[n] = \hat{x}_q[n] + e_q[n]$$

در نتیجه داریم:

$$E(|x[n] - \hat{x}[n]|^2) = E(|e[n] - e_q[n]|^2) \quad (۵-۲)$$

از این روش کوانتیزاسیون هم در سیگنال‌های صوتی و هم تصویر، استفاده شده است.

۲-۱-۲- کدگذاری آنتروپی

حائز اهمیت است که محدودیت‌های تئوری حداقل تعداد بیت‌های لازم برای نمایش یک سیگنال صوتی را در نظر بگیریم. شانون، در "تئوری ریاضی ارتباطات" خود [۳]، نشان داده است که حداقل تعداد بیت‌های لازم برای کدگذاری یک سیگنال مانند X ، با آنتروپی^۱ آن سیگنال، $H_e(X)$ ، مشخص می‌شود.

آنتروپی یک سیگنال دلخواه ورودی می‌تواند به صورت زیر تعریف شود. فرض کنید بردار $X = [x_1, x_2, \dots, x_N]$ بردار ورودی به طول N است به طوری که مقادیر هر کدام از درایه‌های آن از

^۱ Entropy

مجموعه‌ی نمادهای $S = [s_1, s_2, \dots, s_K]$ هستند. همچنین فرض کنید p_i احتمال پیشین این پیشامد است که نماد s_i ارسال شود. اگر n_i تعداد تکرار نماد s_i در بردار X باشد آنگاه p_i از رابطه‌ی (۶-۲) به دست می‌آید:

$$p_i = \frac{n_i}{N} \quad (۶-۲)$$

در این صورت، آنتروپی سیگنال X به صورت (۷-۲) تعریف می‌شود:

$$H_e(X) = - \sum_{i=1}^K p_i \log_2(p_i) \quad (۷-۲)$$

بیشترین آنتروپی زمانی حاصل می‌شود که فرکانس حضور هر کدام از نمادهای s_i در بردار X برابر باشد [۱۰] یعنی $p_1 = p_2 = \dots = p_K$. در این صورت از روی رابطه‌ی (۷-۲)، آنتروپی، برابر $\log_2 K$ خواهد بود. همچنین کمترین میزان آنتروپی، زمانی اتفاق می‌افتد که فقط یک نماد در بردار X داشته باشیم. در این صورت، آنتروپی، صفر خواهد بود. در نتیجه همیشه داریم: $0 \leq H_e(X) \leq \log_2 K$.

به عبارت دیگر آنتروپی، H ، میزان تمرکز هیستوگرام مربوط به نمادهای موجود در سیگنال را مشخص می‌کند [۱۰]. اگر هیستوگرام، یکنواخت باشد، بنا به توضیحات بالا آنتروپی زیاد خواهد بود. اما اگر هیستوگرام تیز باشد، یعنی در اطراف تعداد کمی از نمادها، متمرکز شده باشد، آنتروپی، کم خواهد بود.

به یک رشته‌ی باینری که برای نمایش یک نماد به کار می‌رود، "کلمه‌ی رمز"^۱ گفته می‌شود. طول کلمه‌ی رمز مربوط به نماد s_i را با l_i نشان می‌دهیم. حال بعد از یک کدگذاری باینری سیگنال X ، با فرض اینکه هر نماد s_i در بردار X ، با فرکانس نسبی p_i ظاهر می‌شود، "متوسط طول کلمات رمز" می‌تواند از رابطه‌ی (۸-۲) محاسبه شود:

$$\bar{l} = \sum_{i=1}^K p_i l_i \quad (۸-۲)$$

^۱ Codeword

واحد \bar{L} ، بیت بر نمونه می‌باشد. هر چقدر \bar{L} ، کمتر باشد یعنی فشردگی بیشتر انجام شده است و این مطلوب ماست. شانون [۳] نشان می‌دهد که برای هر نمایش باینری دلخواه از بردار X ، نامساوی زیر برقرار است:

$$\bar{L} \geq H_e(X) \quad (9-2)$$

یعنی آنتروپی محاسبه شده در رابطه‌ی (۷-۲)، یک حد پایین برای متوسط طول کلمات است. بنابراین اگر یک روش کدگذاری باینری، به متوسط طول کلمات رمز تقریباً $H_e(X)$ برسد، بهینه خواهد بود زیرا نمایش کوتاه‌تر کلمات رمز غیر ممکن است. در واقع، هدف از کدگذاری آنتروپی این است که یک نگاهت برگشت پذیر از دنباله‌ی نمادهای ورودی، به یک رشته از بیت‌های b_i داشته باشیم، با این هدف که \bar{L} ، مینیمم شود. باید توجه داشت که برگشت پذیر بودن نگاهت، کافی نیست. بلکه رشته‌ی بیت‌های b_i ، باید طوری باشند که موقع کدگشایی در سمت گیرنده، دقیقاً به همان دنباله‌ی نمادهای اولیه برسیم و حالت دیگری ممکن نباشد [۱۱].

برخلاف روش‌های DPCM و ADPCM^۱ که تزیاید را با بهره گرفتن از همبستگی موجود در نمونه‌های سیگنال حذف می‌کنند، روش‌های کدگذاری آنتروپی، از احتمال وقوع نمادها بهره می‌گیرند [۱۱]. آنتروپی، بیانگر میزان نامعلومی و عدم اطمینان از مقدار یک متغیر تصادفی است [۶]. به عنوان مثال اگر فرکانس حضور نمادها یکسان باشد (هیستوگرام نمادها، یکنواخت باشد)، میزان عدم اطمینان ما از نماد ورودی، زیاد خواهد بود (آنتروپی هم زیاد خواهد بود). اما اگر فرکانس حضور بعضی نمادها زیاد و بعضی کم باشد، در این صورت میزان عدم اطمینان ما نسبت به بردار ورودی، کم خواهد بود (آنتروپی نیز کم خواهد بود).

برای کدگذاری سیگنال‌ها، یک روش ساده این است که اگر تعداد نمادهای موجود در بردار ورودی، در بازه‌ی $2^{B-1} < K \leq 2^B$ باشد، از B بیت برای کد کردن هر نماد استفاده کنیم. اما این

^۱ Adaptive DPCM

روش اصلاً بهینه نمی‌باشد. در کدگذاری آنتروپی، نمادهای ورودی، بر اساس فرکانس حضور هر نماد، تبدیل به کد می‌شوند. مثلاً در روش هافمن، برای نمادهایی که احتمال وقوع زیادتری دارند، تعداد بیت کمی اختصاص داده می‌شود تا تأثیر آن در افزایش نرخ بیت خروجی کم شود. همچنین کدهای (کلمات رمز) با طول بیشتر، به نمادهایی که احتمال وقوع کمتری دارند تخصیص می‌یابند. در این صورت، اگر چه، طول کد بعضی از نمادها که احتمال وقوع کمتری دارند، بیشتر از B می‌شود، اما متوسط طول کلمات رمز، \bar{L} ، کمتر خواهد بود. روش‌های کدگذاری آنتروپی مختلفی ارائه شده‌اند. از آن جمله می‌توان به روش‌های کدگذاری هافمن^۱، حسابی^۲، رایس^۳، گلمب^۴ و Lempel-Ziv-Welch (LZW) [۱۲] اشاره کرد. این روش‌های کدگذاری آنتروپی، روش‌های بدون نویز^۵ نامیده می‌شوند. یک روش کدگذاری بدون نویز قادر است که سیگنال اولیه را به طور کامل از روی نسخه‌ی کد شده‌ی آن بازسازی کند. برعکس، روش‌های کدگذاری که قادر به بازسازی کامل سیگنال نیستند، روش‌های نویزی^۶ نامیده می‌شوند.

در ادامه، دو روش کدگذاری حسابی و دوره‌ی تداوم^۷ را که در این پایان نامه مورد استفاده قرار گرفته‌اند، توضیح می‌دهیم.

۱-۲-۱-۲ - کدگذار حسابی

کدگذار حسابی [۱۳] در مورد کد کردن رشته‌ی نمادهای ورودی به صورت یک رشته‌ی باینری بحث می‌کند. ایده‌ی اصلی در کدگذار حسابی، کد کردن کل بردار ورودی (و نه تک تک نمادها) به صورت یک کلمه‌ی رمز بزرگ است که این کلمه، معادل یک عدد اعشاری در بازه‌ی $[0, 1)$ می‌باشد. بنابراین کدگذار حسابی، برخلاف روش‌هایی مثل هافمن، مبتنی بر کتابخانه نمی‌باشد.

¹ Huffman
² Arithmetic
³ Rice
⁴ Golomb
⁵ Noiseless
⁶ Noisy
⁷ Run-length

کدگذار حسابی مخصوصاً وقتی که با کدگذاری وفقی^۱ سر و کار داریم و همچنین در مواقعی که تعداد نمادهای کم و توزیع نمادهای غیریکنواخت داریم، مناسبتر است [۱۳، ۱۴]. الگوریتم کدگذاری حسابی، به صورت زیر است:

قدم ۱: بازه‌ی جاری: $\text{current-interval} = [0, 1)$

قدم ۲: بازه‌ی جاری را به زیربازه‌هایی با نسبت طول‌های $p_i (i = 1, \dots, K)$ ، افراز کرده و هر نماد s_i را با زیربازه‌ی مربوط به خودش، متناظر می‌کنیم.

قدم ۳: زیربازه‌ی متناظر با نماد ورودی را انتخاب می‌کنیم و بازه‌ی جاری (current-interval) را برابر با زیربازه‌ی انتخاب شده در این مرحله قرار می‌دهیم.

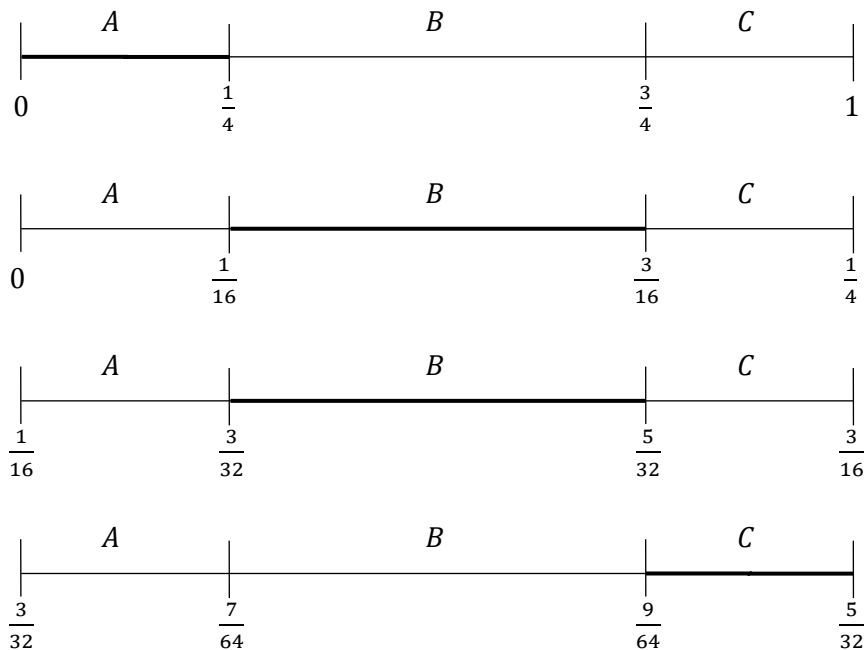
قدم ۴: اگر به آخرین نماد ورودی رسیده‌ایم، الگوریتم را پایان داده و به قدم ۵ می‌رویم. اگر نه، به قدم ۲ می‌رویم.

قدم ۵: در این جا، بازه‌ی جاری را به صورت $[a, b)$ خواهیم داشت. هر عدد باینری که در بازه‌ی مذکور قرار بگیرد، می‌تواند به عنوان رمز، به کار رود و سیگنال ورودی می‌تواند از روی آن به طور کامل بازسازی شود. اما برای رسیدن به مینیمم متوسط طول کلمات، \bar{L} ، عددی را از بازه‌ی مذکور انتخاب می‌کنیم که کمترین طول نمایش باینری را داشته باشد.

برای راحتی درک الگوریتم، یک مثال ساده می‌آوریم. فرض کنید بردار ورودی ما، $X = [ABBC]$ باشد. در این صورت نمادهای ما $S = [A, B, C]$ و احتمالات وقوع متناظر، $p_i = \left[\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right] (i \in S)$ خواهد بود. شکل ۲-۵، مراحل مختلف الگوریتم را برای این مثال نشان می‌دهد. با این توضیح که زیربازه‌هایی که به صورت خط ضخیم رسم شده‌اند، مربوط به زیربازه‌های انتخاب شده در قدم ۳ از الگوریتم می‌باشند. همان طور که از شکل می‌بینیم، آخرین بازه، که در قدم ۵ به آن رسیدیم، بازه‌ی $[a, b) = \left[\frac{9}{64}, \frac{5}{32}\right)$ است. نمایش باینری ابتدای بازه، به صورت

^۱ Adaptive Coding

$a = 0.001001$ و نمایش باینری انتهای بازه، به صورت $b = 0.00101$ می‌باشد. با کمی دقت می‌بینیم که عددی که در بازه‌ی مذکور بوده و کوچکترین نمایش باینری را داشته باشد عدد 0.001001 است. پس خروجی کدگذار حسابی، رشته‌ی باینری 001001 می‌باشد.



شکل ۲-۵. مثال از الگوریتم کدگذاری حسابی

۲-۲-۱-۲ روش رمزگشایی در کدگذار حسابی

با توجه به این که زیربازه‌ها در الگوریتم کدگذاری به صورت سری و تودرتو می‌باشد، نحوه‌ی کدگشایی به صورت زیر است. ابتدا بازه‌ی $[0, 1]$ را به ترتیبی که در بالا گفته شد، به زیربازه‌های مورد نظر، افراز کرده و زیربازه‌ای را که عدد مربوط به کلمه‌ی رمز در آن بازه قرار دارد، انتخاب می‌کنیم. نماد متناظر به این بازه، اولین المان از رشته‌ی نمادهای ورودی (که به دنبال بازسازی آن هستیم) خواهد بود. حال این زیربازه را در نظر گرفته و آن را دوباره، مطابق مرحله‌ی قبل، به زیربازه‌های مورد نظر افراز می‌کنیم. دوباره زیربازه‌ای را که شامل عدد مذکور باشد، انتخاب می‌کنیم تا دومین المان از رشته‌ی نمادهای ورودی نیز بازسازی شود. این کار را ادامه می‌دهیم تا المان‌های بعدی بردار ورودی، بازسازی شوند. توجه داریم که این کار می‌تواند تا بی‌نهایت ادامه پیدا کند. تعداد مراحل رمزگشایی،

برابر تعداد المان‌های بردار ورودی است. بنابراین برای توقف الگوریتم و بازسازی دقیق سیگنال ورودی، باید تعداد المان‌های بردار ورودی را در دست داشته باشیم. واضح است که برای بازسازی سیگنال، احتمال نمادهای ورودی را نیز باید داشته باشیم.

۲-۱-۲-۳ - کدگذاری به روش دوره‌ی تداوم

کدگذاری به روش دوره‌ی تداوم یک روش فشرده‌سازی بدون اتلاف و مؤثر می‌باشد. این روش کدگذاری زمانی مؤثر است که دنباله‌ی ورودی، از قسمت‌هایی متشکل از صفرهای پیوسته، که به دنبال آن‌ها تعداد کمی از نمونه‌های غیرصفر آمده است، تشکیل شده باشد [۱۵]. در این صورت به جای نمایش رشته‌ی صفرها، طول رشته را کد می‌کنیم. برای این کار لازم است تا یک کلمه، به عنوان مشخص‌کننده‌ی شروع رشته‌ی صفرها در نظر بگیریم. یعنی به جای رشته‌ی صفرها، دو کلمه داشته باشیم، یکی، مشخص‌کننده‌ی تکرار صفرها، و دیگری طول رشته‌ی صفرها. مقداری که برای مشخص‌کننده‌ی تکرار به کار می‌رود، نباید در بردار ورودی موجود باشد. بعد از اعمال این روش، به منظور فشرده‌سازی بیشتر، می‌توان هم مقادیر غیر صفر و هم مقادیر مربوط به تکرارهای صفر را با استفاده از روش‌های کدگذاری آنتروپی، کدگذاری نمود.

اگر چه این روش برای تکرار صفرها گفته شد، اما می‌تواند در حالت کلی برای تمام مقادیری که به صورت متوالی تکرار می‌شوند تعمیم داده شود. روشی مشابه با این، در مرجع [۲] برای فشرده‌سازی بدون اتلاف سیگنال‌های گفتار، استفاده شده است. ما نیز برای عملکرد مؤثرتر روش‌های مبتنی بر تبدیل (DCT و DWT)، روش مشابهی با کمی تغییرات، برای ضرایب تبدیل به کار برده‌ایم. جزئیات بیشتر در قسمت مربوطه آمده است.

۲-۲- مروری بر روش‌های فشرده‌سازی با اِتلاف

۲-۲-۱- روش کدینگ سینوسی^۱

نمایش سیگنال‌ها به وسیله‌ی مجموع سینوسی‌ها، و روش‌های تجزیه و ترکیب مبتنی بر این نمایش، تبدیل به یک ابزار ضروری در پردازش موسیقی و صحبت شده است [۱۶]. یکی از مهم‌ترین و مؤثرترین کاربردهای بحث تجزیه و ترکیب سینوسی‌ها، در فشرده‌سازی سیگنال‌های صوت و صحبت است. این بحث به عنوان یکی از روش‌های فشرده‌سازی با نرخ بیت پایین، شناخته شده است. این روش هم برای سیگنال‌های صحبت [۱۷] و هم برای سیگنال‌های موسیقی [۱۸] با موفقیت اعمال شده است. بحث تجزیه و ترکیب سینوسی در اوایل دهه‌ی ۱۹۸۰ شکل گرفت. رابرت مکالی [۱۷] یکی از محققینی است که در این زمینه، تحقیقات زیادی انجام داده است.

پیشنهاد این روش این است که ما می‌توانیم سیگنال‌های صوتی را با استفاده از مجموع تعدادی توابع سینوسی، با فرکانس‌ها، دامنه‌ها و فازهای مختلف، نمایش دهیم. به صورت رابطه‌ی زیر:

$$\hat{s}(n) = \sum_{l=1}^{L(k)} \hat{A}_l^k \cos[n\hat{\omega}_l^k + \hat{\theta}_l^k] \quad (10-2)$$

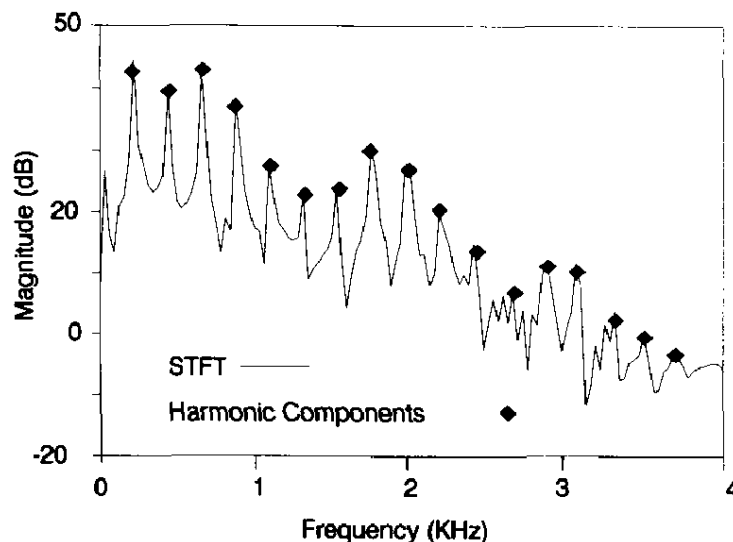
که در آن $\hat{s}(n)$ ، فریم بازسازی شده، و k نشان دهنده‌ی شماره‌ی فریم و l نشان دهنده‌ی شماره‌ی سینوسی در آن فریم، $L(k)$ تعداد سینوسی‌های فریم k ، و نمادهای A ، ω و θ ، به ترتیب نماینده‌ی دامنه، فرکانس و فاز سینوسی هستند. بنابراین چالش پیش رو در این جا، انتخاب و تخمین دقیق‌تر پارامترهای این سینوسی‌ها می‌باشد. روش‌های متفاوتی برای این کار وجود دارد. از جمله‌ی این روش‌ها می‌توان به شناسایی پیک‌ها در تبدیل STFT^۲ [۱۷]، می‌نیم کردن مربع خطا [۱۹] و matching pursuit [۲۰]، اشاره کرد. یکی از معمول‌ترین این روش‌ها، پیدا کردن پیک‌های طیف سیگنال در پریودوگرام^۳ فریم صوتی می‌باشد. مکالی و کواتیری [۱۷] نشان داده‌اند که بازسازی با

^۱ Sinusoidal Coding

^۲ Short Time Fourier Transform

^۳ Periodogram

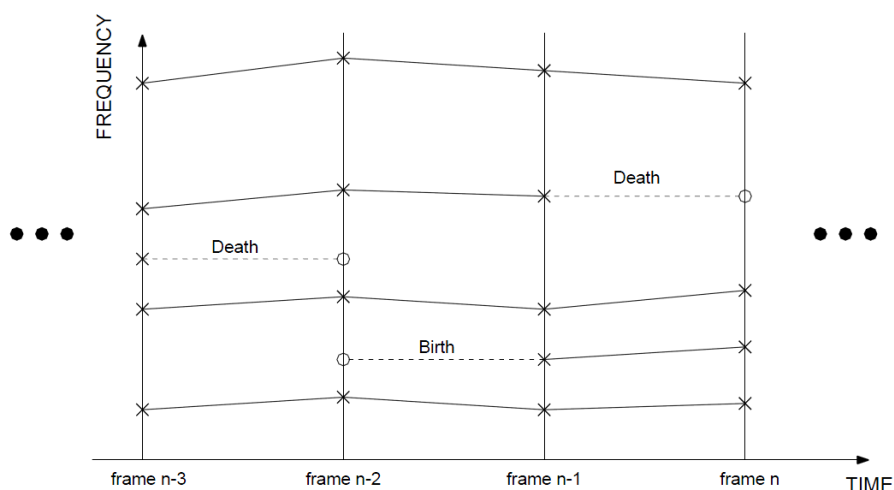
کیفیت بالای سیگنال، می‌تواند با استفاده از سینوسی‌هایی که فرکانس آن‌ها متناظر با پیک‌های تبدیل STFT است، حاصل شود. برای تبدیل STFT، یک پنجره‌ی همینگ با طول $2/5$ برابر دوره‌ی تناوب پیچ^۱ پیشنهاد شده است. در این روش، سیگنال به فریم‌هایی افزای می‌شود و سپس با استفاده از یک پنجره‌ی همینگ، تبدیل STFT گرفته می‌شود. سپس پیک‌های حوزه‌ی تبدیل شناسایی شده و فرکانس‌ها، دامنه‌ها و فازهای متناظر با آن پیک‌ها، استخراج می‌شوند. با استفاده از این دامنه‌ها، فازها و فرکانس‌ها، می‌توان دوباره سیگنال را با استفاده از رابطه‌ی (۲-۱۰) بازسازی کرد. اما این جا مسائلی مطرح می‌شود. اول این که تعداد پیک‌ها در فریم‌ها متفاوت است. هم‌چنین بعضی فریم‌ها تعداد پیک‌های بیش از حدی دارند. علاوه از آن در سیگنال بازسازی شده، در مرز فریم‌ها بریدگی‌هایی به وجود می‌آید و راه حل همه‌ی این‌ها در این مقاله گفته شده است. تعداد پیک‌های انتخاب شده برای هر فریم می‌تواند متفاوت باشد. در هر فریم حداکثر ۸۰ سینوسی مربوط به پیک‌هایی که بیشترین دامنه را دارند انتخاب می‌شوند. شکل ۲-۶ یک مثال از شناسایی پیک‌های STFT را نشان می‌دهد.



شکل ۲-۶. یک مثال از شناسایی پیک‌های تبدیل STFT از روی طیف دامنه

^۱ Pitch

ایده‌ی مهم دیگر این مقاله، ارائه‌ی یک روش برای وصل کردن سینوسی‌های یک فریم، به سینوسی‌های فریم بعدی است. این کار به این خاطر است که از بریدگی‌هایی که به صورت صداهای آزاردهنده به گوش می‌رسند جلوگیری شود. هم‌چنین چون تعداد سینوسی‌ها در فریم‌های مختلف متفاوت است یک الگوریتم «تولد-مرگ»^۱ پیشنهاد شده است. روش وصل کردن سینوسی‌ها در فریم‌های مجاور به این صورت است که اگر فاصله‌ی فرکانس یک سینوسی از فریم جاری، یعنی فریم k ، از فرکانس یک سینوسی دیگر در فریم بعدی (فریم $k + 1$)، از یک آستانه‌ی از قبل مشخص کمتر باشد، این دو سینوسی به هم وصل می‌شوند. حال ممکن است یک سینوسی از فریم k ، به هیچ کدام از سینوسی‌های فریم بعدی وصل نشود. در این صورت، حالت مرگ اتفاق می‌افتد. یعنی این سینوسی را به یک سینوسی فرضی دیگر از فریم $k + 1$ با فرکانس یکسان و دامنه‌ی صفر وصل می‌کنند. اما این سینوسی فرضی نباید در ادامه، برای وصل کردن سینوسی‌ها به یکدیگر مورد استفاده قرار بگیرد. هم‌چنین ممکن است یک سینوسی از فریم $k + 1$ ، با هیچ کدام از سینوسی‌های فریم k ، متصل نباشند. در این صورت حالت تولد اتفاق می‌افتد. یعنی این سینوسی مربوط به فریم $k + 1$ را با یک سینوسی فرضی فریم k با همان فرکانس و با دامنه‌ی صفر، متصل می‌کنیم. یک مثال از وصل کردن سینوسی‌ها و قسمت‌های تولد و مرگ را می‌توانید در شکل ۷-۲ ببینید.



شکل ۷-۲. یک مثال از اتصال داخلی فرکانس‌ها و فرآیند تولد و مرگ

¹ Birth-Death

در شکل ۲-۷ هر علامت "X" نماینده‌ی یک سینوسی تخمین زده شده است و هر علامت ° نماینده‌ی یک سینوسی فرضی است که برای شروع یک اتصال تولد و یا پایان یک اتصال مرگ به کار رفته است.

۲-۱-۱-۱ - سیستم بازسازی

از آنجا که دامنه‌ها، فرکانس‌ها و فازها برای هر فریم تخمین زده شده‌اند، بنابراین معقول به نظر می‌رسد که برای هر فریم، با استفاده از مجموع سینوسی‌ها به صورت رابطه‌ی (۲-۱۰)، آن فریم را بازسازی کنیم. اما مشکل این است که اگر این فریم‌ها به همین شکل بازسازی شوند، به دلیل طبیعت تغییر پذیر با زمان بودن این پارامترها، بریدگی‌هایی در مرز فریم‌ها خواهیم داشت که باعث افت شدید کیفیت سیگنال می‌شود. بنابراین برای رفع این مشکل، در این مقاله پیشنهاد شده است تا بین دامنه‌ها، فرکانس‌ها و فازهای سینوسی‌هایی که به یکدیگر وصل شده‌اند، درون یابی انجام شود. برای درون یابی باید نرم‌ترین تغییر را در نظر گرفت. به همین دلیل برای دامنه‌ها، درون یابی خطی پیشنهاد شده است. زیرا تغییر به صورت خطی، نرم‌ترین تغییر است. بنابراین رابطه‌ی دامنه‌ی هر کدام از نمونه‌های فریم k ، به صورت رابطه‌ی (۲-۱۱) در می‌آید:

$$\tilde{A}(n) = \hat{A}^k + \frac{(\hat{A}^{k+1} - \hat{A}^k)}{S} n \quad (2-11)$$

که در آن $n = 0, 1, \dots, S-1$ ، نمونه‌های زمانی در فریم k هستند. اما متأسفانه برای درون یابی فرکانس‌ها و فازها نمی‌توان از این ایده‌ی ساده استفاده کرد. چرا که فاز اندازه گیری شده به پیمانه‌ی 2π است. یعنی سینوسی‌هایی که فازهایشان اختلاف 2π با هم دارند، در واقع هم فاز هستند. از طرفی، فرکانس، مشتق فاز است. بنابراین هنگام درون یابی فاز، باید آن فازی را در نظر بگیریم که با ازای آن تغییرات فرکانس، بیشترین حالت نرمی را خواهد داشت. اولین قدم برای حل این مشکل این است که برای فاز، یک تابع درون یابی فاز که به صورت یک چندجمله‌ای از مرتبه‌ی ۳ است در نظر می‌گیریم. یعنی:

$$\tilde{\theta}(t) = \zeta + \gamma t + \alpha t^2 + \beta t^3$$

(۱۲-۲)

رابطه‌ی (۱۲-۲) یک چندجمله‌ای از مرتبه‌ی ۳ است که به صورت یک تابع از زمان در نظر گرفته شده است که در لحظه‌ی $t = 0$ متناظر فریم k و در لحظه‌ی $t = T$ متناظر فریم $k + 1$ است. ایده‌ی اعمال یک تابع چندجمله‌ای از درجه‌ی ۳ برای درون‌یابی فاز بین مرزهای فریم‌ها، ابتدا توسط آلمدیا و سیلوا [۲۱] پیشنهاد شد. اما در این مقاله، مکالی و کواتیری، یک روش صریح برای درون‌یابی فاز ارائه داده‌اند که یک محدودیت اضافی برای حل مسأله در نظر می‌گیرد و آن این که تابع درون‌یاب باید بیشترین نرمی را داشته باشد. روابط ریاضی به صورت زیر هستند:

$$\dot{\tilde{\theta}}(t) = \gamma + 2\alpha t + 3\beta t^2 \quad (۱۳-۲)$$

در نقطه‌ی شروع $t = 0$ داریم:

$$\begin{aligned} \tilde{\theta}(0) &= \zeta = \hat{\theta}^k \\ \dot{\tilde{\theta}}(0) &= \gamma = \hat{\omega}^k \end{aligned} \quad (۱۴-۲)$$

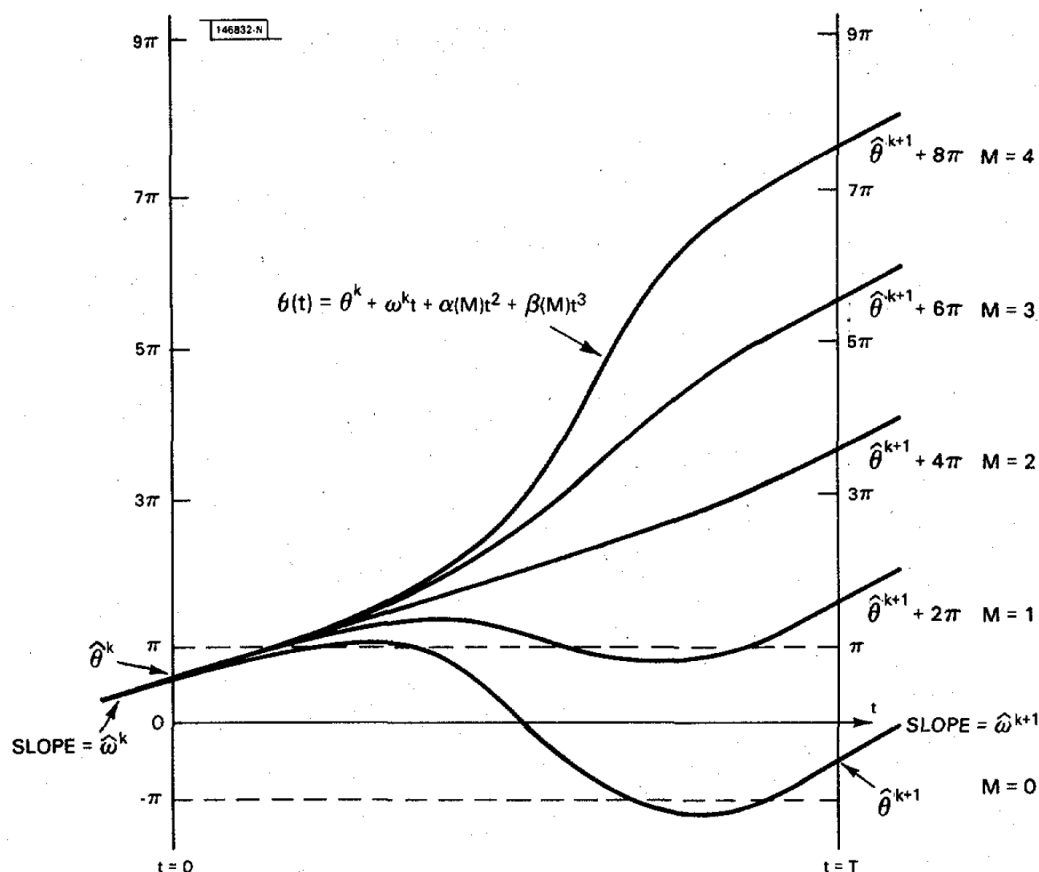
و در نقطه‌ی انتهایی $t = T$ داریم:

$$\begin{aligned} \tilde{\theta}(T) &= \hat{\theta}^k + \hat{\omega}^k T + \alpha T^2 + \beta T^3 = \hat{\theta}^{k+1} + 2\pi M \\ \dot{\tilde{\theta}}(T) &= \hat{\omega}^k + 2\alpha T + 3\beta T^2 = \hat{\omega}^{k+1} \end{aligned} \quad (۱۵-۲)$$

حال در این جا M نامعلوم است. اما مقادیر گسسته دارد و برای هر مقدار M ، هر چه که باشد یک مقدار برای α و β به دست می‌آید. یعنی α و β تابع M هستند و با حل معادله‌ی (۱۶-۲) به دست می‌آیند:

$$\begin{bmatrix} \alpha(M) \\ \beta(M) \end{bmatrix} = \begin{bmatrix} \frac{3}{T^2} & \frac{-1}{T} \\ \frac{-2}{T^3} & \frac{1}{T^2} \end{bmatrix} \begin{bmatrix} \hat{\theta}^{k+1} - \hat{\theta}^k - \hat{\omega}^k T + 2\pi M \\ \hat{\omega}^{k+1} - \hat{\omega}^k \end{bmatrix} \quad (۱۶-۲)$$

برای تعیین M باید معیار بیشترین نرمی تغییر را در نظر بگیریم. شکل ۲-۸ تعدادی از توابع چندجمله‌ای از مرتبه‌ی ۳ را برای مقادیر مختلف M نشان می‌دهد.



شکل ۸-۲. چند نمونه از توابع چندجمله‌ای برای درون‌یابی فاز [۱۷]

واضح است که بهترین تابع فاز، آن است که کمترین تغییرات را داشته باشد. بنابراین یک

معیار منطقی برای "نرمی" این است که M را به گونه‌ای انتخاب کنیم که:

$$f(M) = \int_0^T [\ddot{\theta}(t; M)]^2 dt \quad (17-2)$$

مینیمم شود. علامت $\ddot{\theta}$ به معنی مشتق مرتبه‌ی دوم است. برای این کار می‌توانیم آن را برای $f(x)$ پیوسته حل کرده و نزدیک‌ترین عدد صحیح را به عنوان M انتخاب کنیم. نشان داده می‌شود که x

مطلوب از رابطه‌ی (۱۸-۲) حاصل می‌شود:

$$x^* = \frac{1}{2\pi} \left[(\hat{\theta}^k + \hat{\omega}^k - \hat{\theta}^{k+1}) + (\hat{\omega}^{k+1} - \hat{\omega}^k) \frac{T}{2} \right] \quad (18-2)$$

و از این جا M^* به دست می‌آید و بنابراین با محاسبه‌ی $\alpha(M^*)$ و $\beta(M^*)$ از روی رابطه‌ی (۱۶-۲)،

تابع درون‌یاب فاز به صورت (۱۹-۲) حاصل می‌شود:

$$\tilde{\theta}(t) = \hat{\theta}^k + \hat{\omega}^k t + \alpha(M^*)t^2 + \beta(M^*)t^3 \quad (19-2)$$

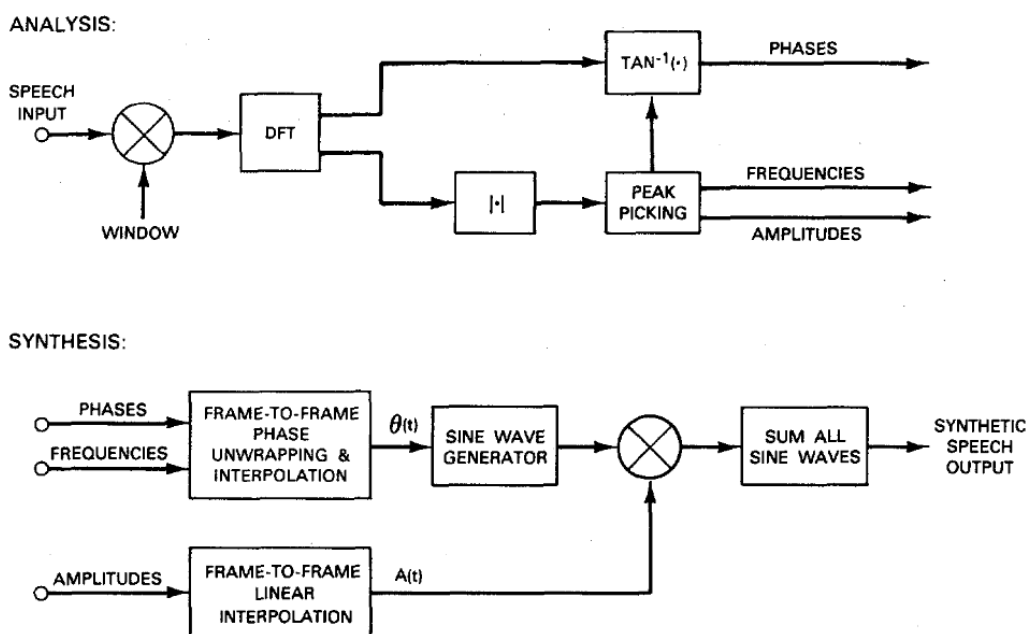
بنابراین سیگنال بازسازی شدهی نهایی، به صورت (۲۰-۲) است:

$$\tilde{s}(n) = \sum_{l=1}^{L(k)} \tilde{A}_l(n) \cos[\tilde{\theta}_l(n)] \quad (20-2)$$

که در آن $\tilde{A}_l(n)$ از رابطهی (۱۱-۲) محاسبه می‌شود. این جا تئوری بحث تجزیه و ساخت سینوسی به پایان می‌رسد. بلوک دیاگرام روش گفته شده در شکل ۹-۲ آمده است.

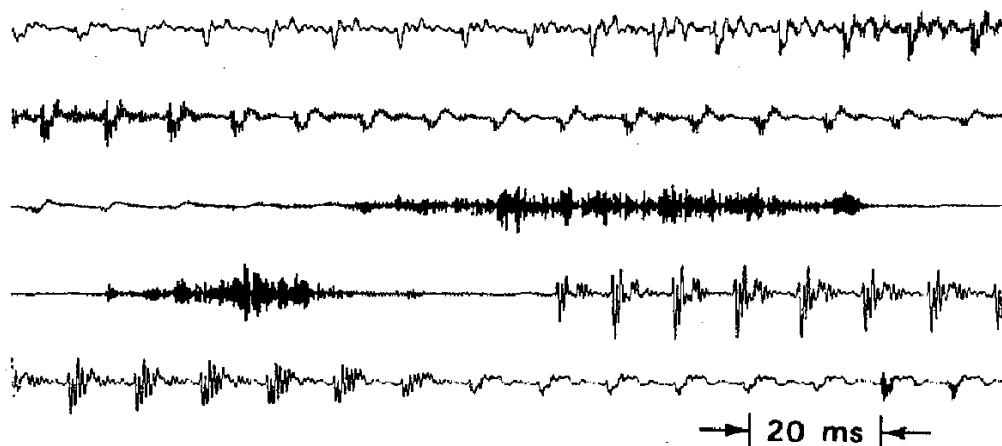
۲-۱-۲-۲ نتایج روش تجزیه و ساخت سینوسی

نتایج این روش [۱۷] نشان می‌دهد که سیگنال بازسازی شده اگر چه عیناً مثل شکل موج سیگنال اصلی نیست اما بسیار مشابه به آن می‌باشد. سیگنال‌های بازسازی شده از لحاظ شنوایی و ادراکی با سیگنال اصلی تقریباً غیرقابل تشخیص هستند. این روش ارائه شده می‌تواند برای تمام انواع سیگنال‌ها و شکل موج‌هایی که از مجموع شکل موج‌های سینوسی با دامنه‌ها و فرکانس‌های تغییرپذیر با



شکل ۹-۲. بلوک دیاگرام سیستم تجزیه و ساخت سینوسی [۱۷]

تشکیل شده‌اند، کاربرد داشته باشد. این نظریه با آزمایش انواع سیگنال‌های مختلف، شکل موج‌های چندصحت‌کننده، موسیقی، صحبت همراه موسیقی، صداهاى دریایی مثل صدای نهنگ، با موفقیت تأیید شد. علاوه از آن، در حضور نویز نیز، بازسازی انجام می‌شود. یعنی سیگنال بازسازی شده، با سیگنال نویزی اصلی، تقریباً غیرقابل تشخیص است و مشخصات نویز تقریباً بدون تغییر باقی می‌ماند. شکل ۱۰-۲ چند مثال از سیگنال‌های اصلی و بازسازی شده آنها را نشان می‌دهد.



(الف)



(ب)

شکل ۱۰-۲. سیگنال بازسازی شده از روش سینوسی [۱۷]. الف) سیگنال اصلی. ب) سیگنال بازسازی شده
 یک نظریه وجود دارد که می‌گوید گوش انسان نسبت به فاز حساس نیست. پیشنهادی که پایه‌ی خیلی از کارهای انجام شده در زمینه‌ی فشرده‌سازهای صوت محسوب می‌شود. یک آزمایش در این زمینه انجام شده که در آن، هر تابع چندجمله‌ای درجه ۳ برای درون یاب فاز، با یک تابع فاز که

انتگرال فرکانس‌های لحظه‌ای است جاگذاری شده است [۲۲]. این روش بازسازی مبتنی بر فقط دامنه، به چندین سیگنال صحبت اعمال شده است و نتایج نشان می‌دهد که هرچند سیگنال بازسازی شده، کاملاً واضح است اما متفاوت از سیگنال اصلی به گوش می‌رسد. آزمایش‌های سیستم مبتنی بر فقط دامنه، بر روی تعدادی سیگنال نویزی نیز امتحان شد که نتایج نشان می‌دهد، نویز بازسازی شده، یک کیفیت تونلی داشته و غیرطبیعی و آزاردهنده است.

مکالری و کواتیری در یک کار دیگر، با استفاده از روش تجزیه و ساخت خود، یک فشرده‌ساز صحبت طراحی کردند که در نرخ بیت ۸ کیلوبیت بر ثانیه عمل می‌کند [۲۳]. این فشرده‌ساز براساس سیستم تجزیه و ساخت سینوسی که گفته شد، طراحی شده است. فرکانس فریم‌ها در این فشرده‌ساز، ۵۰ هرتز و $3/6\text{ kbit/s}$ برای کدگذاری دامنه‌ها و $4/4\text{ kbit/s}$ برای کدگذاری فازها و فرکانس‌های پایه به کار رفته است. دامنه‌های سینوسی‌ها، با استفاده از روش DM^۱ انجام شده است که توضیحات آن در بخش مربوط به کوانتیزاسیون آمده است.

۲-۲-۲- روش انطباق الگو

در روش انطباق الگو^۲، سعی بر این است که از شباهت‌های موجود در قسمت‌های مختلف سیگنال، استفاده کرده و به کاهش حجم سیگنال کمک کرد. در این روش، یک الگو از بلوک‌های مشابهی که رخداد زیادی دارند، در یک کتابخانه ذخیره می‌شوند و الگوهای بعدی، از سیگنال حذف شده و به جای آن‌ها اشاره‌گرهایی به الگوهای موجود در کتابخانه، قرار داده می‌شود. این روش را می‌توان هم بر روی سیگنال در حوزه‌ی زمان و هم بر روی پارامترهای استخراج شده از سیگنال اعمال کرد. از این روش به وفور در فشرده‌سازی سیگنال‌های صوت و تصویر استفاده شده است. یکی از معروف‌ترین و معمول‌ترین این روش‌ها چندی‌سازی برداری^۳ [۷] می‌باشد که در آن تعدادی بردار را به عنوان

^۱ Delta Modulation

^۲ Pattern Matching

^۳ Vector Quantization

بردارهای کتابخانه^۱ در نظر می‌گیرند. سپس هر کدام از بردارهای ورودی، به یکی از این بردارهای الگو، نگاشت می‌شوند. هر چه تعداد این بردارهای الگو، بیشتر باشد، دقت چندی‌سازی بهتر خواهد بود. پیدا کردن محل مناسب این بردارهای الگو، مهم بوده و در نتیجه‌ی فشرده‌ساز تأثیر دارد. چندی‌سازهای برداری انواع مختلفی دارند که در این جا آن‌ها را بحث نمی‌کنیم. در ادامه یک روش نسبتاً جدید [۲۴] که برای فشرده‌سازی سیگنال‌های موسیقی ارائه شده است را مرور می‌کنیم.

بنا به این واقعیت که موسیقی و به ویژه صداها، ایجاد شده از منابع الکترونیکی در یک قطعه‌ی موسیقی، دارای تکرارهای ذاتی است، در این مقاله الگوریتم فشرده‌سازی پایه‌ی Lempel-Ziv [۱۲] تعمیم داده شده است تا یک پنجره از صوت را با استفاده از ترکیب پنجره‌های فیلتر شده‌ی گذشته، نمایندگی کرد. تکرار یک بخش اساسی موسیقی است و پیامد طبیعی این واقعیت است که ابزار و صداها و تن‌های مجزا، برای ایجاد یک قطعه‌ی صوتی استفاده می‌شوند.

در مورد سیگنال‌های صوتی و موسیقی ممکن است بعضی بلوک‌ها از لحاظ اقلیدسی کاملاً متفاوت باشند اما سیستم شنیداری انسان، آنها را مساوی باهم دریافت کند. با وجود این، موسیقی الکترونیکی معمولاً از جایگزینی دقیق و یا ترکیب بلوک‌های مشخص صوتی تولید می‌شود. در یک مدل با حافظه، این ایده می‌تواند بررسی شود تا نرخ فشرده‌سازی روش‌های بدون حافظه‌ی قبلی را بهبود ببخشد.

در این مقاله ابتدا یک بررسی آماری روی سیگنال‌های پایگاه داده، انجام شده است و نشان داده شده که در سیگنال‌های موسیقی برای یک بلوک از سیگنال، احتمال حضور حداقل یک بلوک مشابه با همبستگی بالا در شصت ثانیه‌ی بعدی سیگنال زیاد است. برای جزئیات بیشتر در مورد این بررسی آماری می‌توانید به [۲۴] مراجعه کنید.

ایده اصلی در روش پیشنهادی [۲۴]، براساس تعمیم روش فشرده‌سازی مبتنی بر پیش‌بینی^۲، و الگوریتم فشرده‌سازی Lempel-Ziv است. روش پیش‌بین خطی یک عملیات ریاضی است که در آن

¹ Prototypes/Codebook

² Predictive Coding

مقدار فعلی یک سیگنال گسسته در زمان به صورت تابع خطی از نمونه‌های قبلی (پیش‌بین پیش‌رونده^۱) و یا نمونه‌های آینده (پیش‌بین پس‌رونده^۲) تخمین زده می‌شود. الگوریتم LZ77 از Lempel-Ziv [۱۲] یک کتابخانه یا تاریخچه از داده‌هایی که بیشترین تعداد مشاهدات اخیر را داشته‌اند در نظر گرفته و داده‌ی فعلی ورودی را با داده‌های موجود در پنجره‌ی تاریخچه مقایسه می‌کند. چیزی که در رشته‌ی خروجی ذخیره می‌شود، ارجاعات به محل این داده‌ها در پنجره‌ی تاریخچه، و هم‌چنین طول این داده‌ها است. اگر یک مطابقت بین داده‌ی ورودی و داده‌های پنجره‌ی تاریخچه یافت نشد، آن قسمت مستقیماً در رشته‌ی خروجی کد می‌شود و یک پرچم هم مشخص می‌کند که این قسمت از رشته، داده‌ی واقعی است. الگوریتم‌های گوناگونی از این روش در کاربردهای فشرده‌سازی عمومی و بدون اتلاف استفاده شده است.

در این مقاله، الگوریتم LZ77، برای یک فشرده‌سازی با اتلاف تعمیم داده شده است. هر بلوک سیگنال که می‌تواند از طول‌های مختلفی باشد، به صورت ترکیب خطی از بلوک‌های موجود در کتابخانه به همراه یک بردار خطا نمایندگی می‌شود. ترکیب خطی از بلوک‌های پیشین، با استفاده از اشاره‌گرهای به محل آنها در پنجره‌ی تاریخچه کد می‌شوند. بردار خطا با استفاده از مدل‌های ادراکی^۳، کد می‌شود. هدف این است که اتلاف اطلاعات انجام داده شود تا این که آنتروپی بردار خطا می‌نیم شود. اما این اتلاف اطلاعات باید طوری باشد که از لحاظ ادراک شنوایی قابل تحمل باشد. یعنی اتلاف بر روی قسمت‌هایی که از لحاظ درک شنوایی انسان اهمیت کمتری داشته باشند انجام شود.

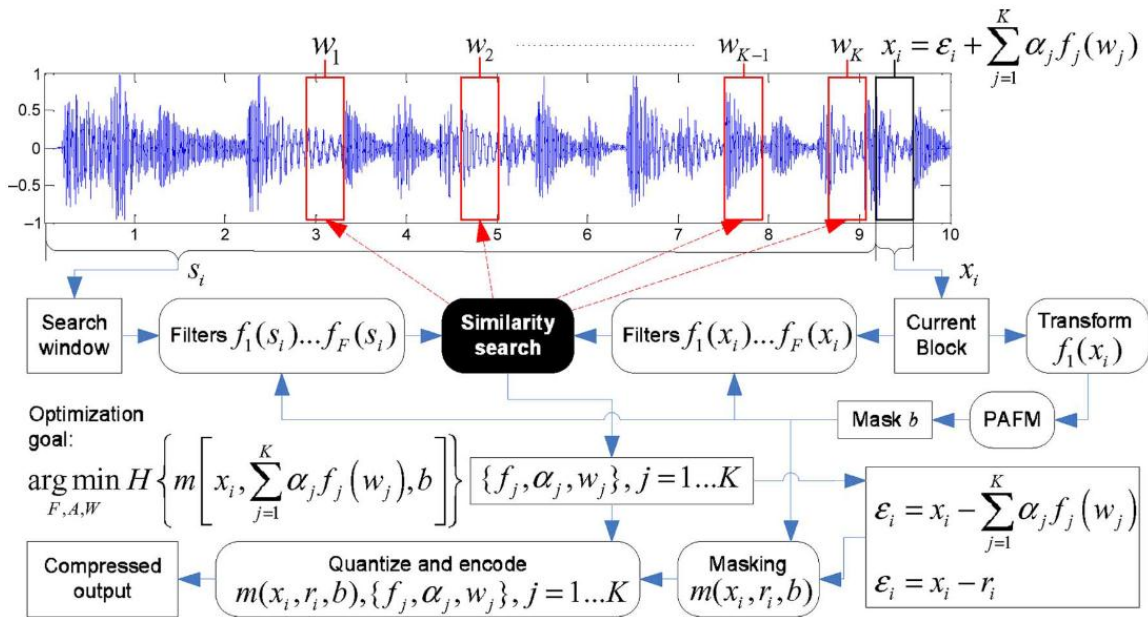
ابتدا سیگنال ورودی نرمالیزه شده‌ی $x \in R^N$ با N نمونه را داریم که هر نمونه‌ی $x_i \in x$ نرمالیزه شده است، یعنی $x_i \in [-1,1]$. سیگنال به بلوک‌هایی با 50% هم پوشانی بخش بندی می‌شود. هر بلوک، n نمونه دارد که n توانی از ۲ است و معمولاً $n \in [512,4096]$. ما یک بلوک از سیگنال با n نمونه را به صورت $x_i = \{x_i, \dots, x_{i+n-1}\}$ نشان می‌دهیم. در طول بازسازی سیگنال، عکس عملیات

¹ Forward Prediction

² Backward Prediction

³ Perceptual Model

تجزیه انجام می‌شود. در این مقاله تبدیل MLT^۱ به عنوان یک تبدیل فرکانسی در نظر گرفته شده است، $\psi(R^n) \rightarrow C^n$ [۲۵]. از آن جا که این تبدیل می‌تواند سیگنال حوزه‌ی زمان را فقط از روی مقادیر حقیقی‌اش به طور کامل بازسازی کند، بنابراین بازسازی را با تابع $f: R^n \rightarrow R^n$ نشان می‌دهیم. در نهایت یک تابع ماسک گذاری صوتی- ادراکی^۲، $\phi(\psi(R^n)) \rightarrow \{0,1\}^n$ به سیگنال در حوزه‌ی فرکانس اعمال می‌شود. خروجی این تابع یک رشته‌ی باینری است که قابلیت استماع ضرایب فرکانس را مشخص می‌کند. ماسک صوتی- ادراکی x_i با b_i نشان داده شده است. اگر ضرایب z از b_i برابر یک باشد یعنی z ضریب از $f(x_i)$ قابل شنیدن است و برعکس. این مدل، براساس یک آستانه‌ی شنیداری T که برای گروه‌های خاصی از ضرایب MLT محاسبه شده‌اند، تصمیم گیری می‌کند. شکل ۲-۱۱، بلوک دیاگرام الگوریتم پیشنهادی این مقاله را نشان می‌دهد.



شکل ۲-۱۱. بلوک دیاگرام روش پیشنهادی در [۲۴]

روند کار به این صورت که برای بلوک ورودی x_i ، یک پنجره‌ی جست و جوی s_i در نظر گرفته می‌شود. که می‌تواند از ابتدای سیگنال اصلی تا بلوک x_i باشد و یا این که یک طول مشخص مانند S

¹ Modulated Lapped Transform

² Psychoacoustic

داشته باشد. در داخل این پنجره، جست و جو برای یافتن یک مجموعه K تایی از بلوک‌های به طول $W = \{w_j \in s_i, j = 1, \dots, K\}$ و یک مجموعه‌ی K تایی از ضرایب اسکالر متناظر، $F = \{f_j: R^n \rightarrow R^n, f_j \in \mathbb{F}, j = 1 \dots K\}$ و تبدیل‌های متناظر، $A = \{a_j \in R, j = 1, \dots, K\}$ انجام می‌شود که در آن \mathbb{F} ، مجموعه‌ی تبدیل‌های در نظر گرفته شده است. پارامتر K می‌تواند برای هر بلوک، متفاوت باشد. تبدیل‌های مورد استفاده در این جا، تبدیل MLT، MDCT و غیره است.

بلوک x_i ورودی با استفاده از یک پیش بین خطی به صورت زیر مدل می‌شود:

$$r_i = \sum_{j=1}^K a_j f_j(w_j) \quad (21-2)$$

سیگنال خطا، $\varepsilon_i = x_i - \sum_{j=1}^K a_j f_j(w_j) = x_i - r_i$ ، با استفاده از یک فیلتر صوتی- ادراکی $m()$ به صورت زیر ماسک گذاری می‌شود. ابتدا ماسک صوتی-ادراکی b_i را برای بلوک x_i محاسبه می‌کنیم. حال تابع ماسک گذاری $m(x, r, b)$ را بر روی یک ضریب منفرد سیگنال، x ، و نمونه بازسازی شده‌ی آن، r و بیت ماسک گذاری b به صورت زیر تعریف می‌کنیم:

$$m(x, r, b) \equiv \begin{cases} x - r & b = 1 \\ 0 & b = 0 \wedge |r| \leq T \\ r - T \cdot \text{sign}(r) & b = 0 \wedge |r| > T \end{cases} \quad (22-2)$$

که در آن T ، آستانه‌ی شنوایی برای نمونه‌ی x است. هنگام اعمال به بردارها، هر تابع $m(x, r, b)$ رابطه‌ی (22-2) را به هر کدام از المان‌های بردارها، به صورت مستقل اعمال می‌کند. هدف تابع $m()$ این است که خطا را به شکلی در نظر بگیرد که بازسازی نمونه‌های قابل شنیدن، به صورت دقیق و بدون خطا باشد، در حالی که بازسازی نمونه‌های غیرقابل استماع، طوری باشد که قدرمطلق دامنه‌ی خطا می‌نیمم شود. بدون در نظر گرفتن این ایده، بهبود روش پیشنهادی در این مقاله، ناچیز است [24].

مجموعه‌های \mathbb{F} ، A ، W در طول جست و جوی شباهت، با هدف بهینه کردن عبارت (23-2)، انتخاب می‌شوند.

$$\arg \min_{W,A,\mathbb{F}} H \left\{ m \left[\mathbf{x}_i, \sum_{j=1}^K a_j f_j(\mathbf{w}_j), \mathbf{b}_i \right] \right\} \quad (23-2)$$

که در آن تابع $H(\cdot)$ ، آنتروپی اطلاعات زیر را اندازه می‌گیرد:

H1. اشاره گرهای کد شده به تمامی بلوک‌های مجموعه‌ی W .

H2. اشاره گرهای کد شده به تبدیل‌های اعمال شده \mathbb{F} .

H3. اعداد اسکالر کد شده در مجموعه‌ی A که برای ایجاد ترکیب خطی از بلوک‌های تبدیل

استفاده می‌شوند.

H4. بردار خطای کد شده که با تابع $m(\cdot)$ بدست آمده است.

واضح است که هدف بهینه سازی در جست و جوی شباهت در رابطه‌ی (23-2)، پیدا کردن مجموعه‌ی W از K بلوک است که قبل از \mathbf{x}_i حضور دارند و همچنین پیدا کردن ترکیب خطی از تبدیل آنها، \mathbf{r}_i ، است به طوری که نزدیک‌ترین به \mathbf{x}_i باشد و آنتروپی خطای باقی مانده $m(\mathbf{x}_i, \mathbf{r}_i, \mathbf{b}_i)$ می‌نیمم شود.

۲-۲-۱- بازسازی سیگنال کد شده

اگرچه فرآیند فشرده‌سازی الگوریتم پیشنهادی، محاسبات بسیار بالایی دارد، قسمت بازسازی آن، ساده‌تر می‌باشد. بازسازی سیگنال را می‌توان به دو بخش یا گام تقسیم کرد. در گام اول، اطلاعات مربوط به رشته‌ی فشرده شده را رمزگشایی می‌کنیم: مجموعه‌ی اشاره‌گرها به بلوک‌های پیشین W را استخراج می‌کنیم. همچنین، اعداد اسکالر A و فیلترهای \mathbb{F} اعمال شده به W ، و بردار خطای $\varepsilon_i = m(\mathbf{x}_i, \mathbf{r}_i, \mathbf{b}_i)$ را استخراج می‌کنیم. در گام آخر، سیگنال بازسازی شده را با رابطه‌ی $\mathbf{x}_i = \sum_{j=1}^k a_j f_j(\mathbf{w}_j) + \varepsilon_i$ به دست می‌آوریم.

۲-۳- مروری بر روش‌های بدون اتلاف

روند اغلب فشرده‌سازهای بدون اتلاف به این صورت است که ابتدا در سیگنال ورودی بعد از فریم بندی، با استفاده از ابزارهای متفاوت، همبستگی و تزیاید موجود در سیگنال را حذف می‌کنند تا حجم سیگنال کاهش یابد. سپس با استفاده از کدگذارهای آنتروپی، نرخ بیت را هرچه بیشتر کاهش می‌دهند. الگوریتم‌های مختلف از روش‌های متفاوتی برای حذف همبستگی استفاده می‌کنند. در ادامه، سه روش فشرده‌سازی بدون اتلاف را معرفی می‌کنیم.

۲-۳-۱- روش Audio Pak

روش Audio Pak [۲۶]، به ویژه در نقل و انتقالات اینترنتی مناسب است زیرا حجم محاسباتی کم و عملکرد فشرده‌سازی بهتری دارد. در این روش از چندجمله‌ای‌ها برای پیش بینی نمونه‌ها استفاده می‌شود. جزئیات روش به صورت زیر است:

۲-۳-۱-۱- پنجره بندی

همانند اغلب روش‌های فشرده‌سازی بدون اتلاف، در این جا نیز سیگنال ورودی به پنجره‌های مستقل افراز می‌شود. طول فریم یکی از پارامترهای این روش بوده و در زمان کدگذاری در رشته‌ی خروجی نوشته می‌شود. با وجود این، دلایل عملی و تجربی وجود دارد که پیشنهاد می‌شود طول فریم‌ها، مضربی از ۱۹۲ باشند. این عدد از طرف AES^۱ و EBU^۲ پیشنهاد شده است. جدول ۱-۲ نتایج این روش را برای پنجره‌های با طول مختلف که مضربی از ۱۹۲ هستند نشان می‌دهد. سیگنال‌ها از پایگاه داده‌ی [۲۷] می‌باشند. از این جدول مشاهده می‌شود که طول فریم ۱۱۵۲ مناسب است.

^۱ Audio Engineering Society

^۲ European Broadcasting Society

جدول ۱-۲. نرخ فشرده‌سازی برای کانال چپ فایل‌های صوتی از [۲۷] در طول فریم‌های مختلف

سیگنال	۱۹۲	۵۷۶	۱۱۵۲	۲۳۰۴	۴۶۰۸
Track 04	۱/۳۸	۱/۳۹	۱/۳۹	۱/۳۸	۱/۳۶
Track 20	۳/۰۲	۳/۱۴	۳/۱۷	۳/۱۷	۳/۱۷
Track 27	۲/۵۰	۲/۴۹	۲/۴۶	۲/۴۲	۲/۳۷
Track 48	۲/۴۸	۲/۵۵	۲/۵۶	۲/۵۶	۲/۵۶
Track 66	۲/۴۲	۲/۴۷	۲/۴۷	۲/۴۶	۲/۴۲

۲-۱-۳-۲ حذف همبستگی

در این روش عملیات حذف همبستگی داخلی، از روش پیش بینی فوقی FIR استفاده می‌کند. این روش پیش بینی از چهار پیش بین FIR استفاده می‌کند. هر کدام از این پیش بین‌ها فقط ضرایب صحیح دارند. این چهار پیش بین به صورت زیر است:

$$\begin{cases} \hat{x}_0[n] = 0 \\ \hat{x}_1[n] = x[n-1] \\ \hat{x}_2[n] = 2x[n-1] - x[n-2] \\ \hat{x}_3[n] = 3x[n-1] - 3x[n-2] + x[n-3]. \end{cases} \quad (24-2)$$

این پیش بین‌ها، پیش بین‌های FIR هستند زیرا فقط ترکیبات خطی تعداد محدودی از نمونه‌های قبلی هستند. یک ویژگی جالب این تقریب‌های چندجمله‌ای آن است که، سیگنال خطای هر کدام از آن‌ها، $e_p[n] = x[n] - \hat{x}_p[n]$ ، می‌تواند با استفاده از روش بازگشتی (۲-۲۵) محاسبه شود:

$$\begin{cases} e_0[n] = x[n] \\ e_1[n] = e_0[n] - e_0[n-1] \\ e_2[n] = e_1[n] - e_1[n-1] \\ e_3[n] = e_2[n] - e_2[n-1] \end{cases} \quad (25-2)$$

این، اجازه می‌دهد تا تمامی سیگنال‌های خطا، بدون هیچ عمل ضربی، محاسبه شوند. این عملیات را می‌توان با استفاده از معماری MMX SIMD همان طور که در [۲۶] نشان داده شده است به صورت موازی انجام داد.

برای هر نمونه از یک فریم، هر چهار خطای مذکور محاسبه می‌شود و قدرمطلق این خطاها برای تمام نمونه‌های فریم با هم جمع می‌شوند. سیگنال خطایی که مجموع قدرمطلق مقادیر آن نسبت به سه مورد دیگر، کمتر باشد به عنوان پیش بین بهینه برای آن فریم انتخاب می‌شود. برای مشخص کردن این که کدام پیش بین برای یک فریم انتخاب شده است از دو بیت استفاده می‌شود که این دو بیت هم در میان اطلاعات مربوط به آن فریم ذخیره می‌شود. جدول ۲-۲ نشان می‌دهد که هر کدام از این پیش بین‌ها چندبار برای سیگنال‌های پایگاه داده استفاده شده‌اند. همان طور که دیده می‌شود هر کدام از این پیش بین‌ها به تعداد قابل توجهی استفاده می‌شوند.

جدول ۲-۲. تعداد دفعاتی که هر کدام از سیگنال‌های خطا، انتخاب شده‌اند.

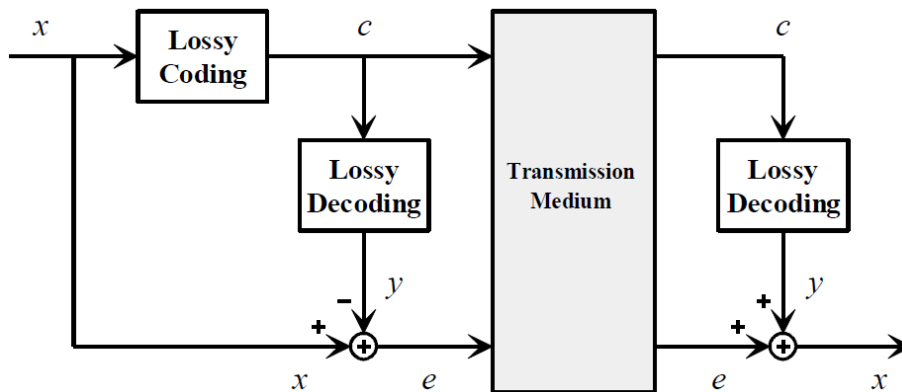
سیگنال	e_0	e_1	e_2	e_3	کل فریم‌ها
Track 04	۸۰	۴۹۴۸	۵۶۹۴	۳۹۱۲	۱۴۶۳۴
Track 20	۶۲۱	۳۶۴	۱۳۰۰	۷۰۱	۲۹۸۶
Track 27	۳۳۴	۳۰۰	۶۶۶	۲۳۲	۱۵۳۲
Track 48	۲۸۸	۲۵۲	۳۳۷	۱۲۶۷	۲۱۴۴
Track 66	۳۰۱	۳۰	۴۰۷	۶۴۱	۱۳۷۹

۲-۳-۱-۳- کدگذاری آنتروپی

فریم‌های سکوت می‌تواند با استفاده از خطاهای $e_0[n]$ و $e_1[n]$ شناسایی شده و به راحتی کد شوند. اگر مقادیر فریم سکوت صفر باشند، مجموع $|e_0[n]|$ صفر خواهد بود و اگر فریم سکوت از مقادیری غیر از صفر باشد (مثلاً ۱ یا -۱) که بعضی وقت‌ها دیده می‌شوند، در این صورت مجموع $|e_1[n]|$ صفر خواهد بود، زیرا $e_1[n] = x[n] - x[n-1]$. اگر فریم یک مقدار ثابت نداشته باشد، از روش کدگذاری Golomb استفاده می‌شود.

۲-۳-۲- روش LTAC

روش LTAC^۱ [۲۸]، یک روش مبتنی بر تبدیل می‌باشد. در حالت کلی، مقادیر حاصل از تبدیل یک سیگنال، مقادیر حقیقی هستند. برای فشرده‌سازی مؤثر، لازم است تا این ضرایب کوانتیزه شوند که به ناچار موجب اتلاف می‌شود. بنابراین یک فشرده‌سازی مبتنی بر تبدیل بدون اتلاف، باید به صورت ترکیبی از فشرده‌سازی مبتنی بر تبدیل با اتلاف و ارسال خطای حاصل از آن باشد [۲۹]. شکل ۱۲-۲ شمای کلی این فشرده‌سازی را نشان می‌دهد. سیگنال c ، از فشرده‌سازی با اتلاف سیگنال اصلی به دست می‌آید. در محل کدگذاری، y را از روی c بازسازی می‌کنیم و خطای بین y و سیگنال اصلی (یعنی e) را نیز همراه c ارسال می‌کنیم.



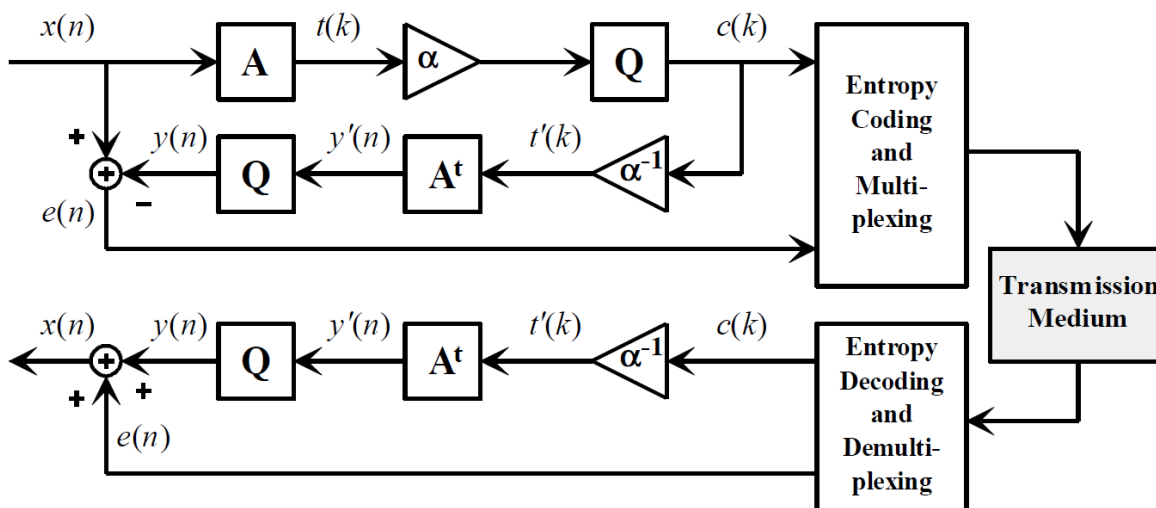
شکل ۱۲-۲. فشرده‌سازی بدون اتلاف، به عنوان ترکیبی از فشرده‌سازی با اتلاف و ارسال سیگنال خطا

مشخص است که سیگنال خطا، شدیداً به روش فشرده‌سازی با اتلاف بستگی دارد. اگر فشرده‌سازی زیاد انجام شود، سیگنال خطا بزرگ و همبسته خواهد بود. از طرف دیگر، اگر فشرده‌سازی خیلی کم باشد، خطا صفر خواهد بود، اما تمام تزیاید موجود در سیگنال حذف نخواهد شد. بنابراین ما باید بهترین و بهینه‌ترین را انتخاب کنیم.

شکل ۱۳-۲ بلوک دیاگرام روش ارائه شده در [۲۸] را نشان می‌دهد. ابتدا سیگنال کوانتیزه شده $X(n)$ ، که مقادیر آن صحیح است، در طول یک تبدیل اورتونرمال A ، ضرایب تبدیل

^۱ Lossless Transform Audio Coding

$t(k)$ را نتیجه می‌دهد. ضرایب با ضریب α تغییر مقیاس داده شده و با یک کوانتیزه کننده، به مقادیر صحیح کوانتیزه می‌شوند ($c(k)$)، که این مقادیر بعد از کدگذاری آنروپی، ارسال می‌شوند. به دلیل این که کوانتیزاسیون انجام شده بود، مقادیر $c(k)$ نمی‌توانند سیگنال اصلی را بدون خطا بازسازی کنند. بنابراین در قسمت کدگذاری، $c(k)$ را کدگشایی می‌کنیم تا سیگنال خطا را به دست آوریم.

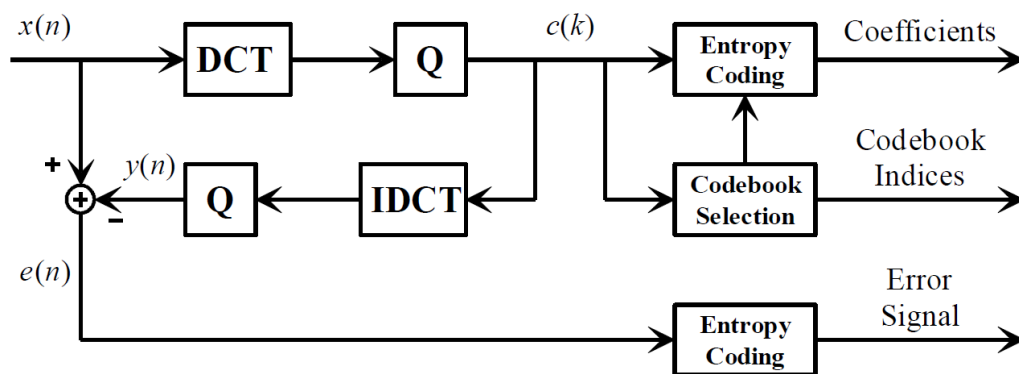


شکل ۲-۱۳. بلوک دیاگرام سیستم فشرده‌سازی بدون اتلاف مبتنی بر تبدیل [۲۸]

برای کدگشایی $c(k)$ ، ابتدا آن را با ضریب a^{-1} دوباره تغییر مقیاس داده و عکس تبدیل قبلی را اعمال می‌کنیم. در این صورت مقادیر حقیقی برای $y'[n]$ حاصل می‌شوند که با کوانتیزه کردن آن مقادیر صحیح $y(k)$ را خواهیم داشت. حال اختلاف بین سیگنال اصلی و $y(k)$ را هم همراه $c(k)$ ارسال می‌کنیم.

در این روش، فریم‌ها می‌توانند با طول ثابت یا وفقی باشند. در حالت طول ثابت فریم‌ها، هر فریم به طول N ، به وسیله‌ی تبدیل DCT با همان طول تبدیل می‌شود. در حالت طول وفقی فریم‌ها، سیگنال ورودی به فریم‌هایی با تعداد نمونه‌های 2048 فریم بندی می‌شود. هر ترکیب از تبدیل‌های 2048 و 1024 و 512 نقطه‌ای محاسبه می‌شوند و برای این فریم، نهایتاً مناسب‌ترین ترکیب انتخاب می‌شود.

آزمایشات با تبدیل‌های مختلف مانند DFT, DST و هارتلی گسسته نشان دهنده‌ی برتری تبدیل DCT نسبت به آن هاست. تبدیل دیگری نیز مانند LOT, MLT, آزمایش شدند. در این تبدیلات در بهترین حالت میزان نرخ بیت به اندازه‌ی ۰/۱ بیت بر نمونه بهبود داشته اما این بهبود اندک، در ازای افزایش حجم محاسبات و از دست دادن قابلیت تصحیح و تغییر بخش‌های مجزای سیگنال بوده است. به همین دلیل در این مقاله، تبدیل DCT انتخاب شده است. بلوک دیاگرام این روش در شکل ۲-۱۴ نشان داده شده است. با توجه به این که توزیع مقادیر $c(k)$ به صورت لاپلاسیان است، بنابراین از کدگذاری rice-coding استفاده می‌شود.



شکل ۲-۱۴. بلوک دیاگرام فشرده‌ساز بدون اتلاف مبتنی بر تبدیل (LTAC)

این مقاله نتایج خود را با روش بدون اتلاف SHORTEEN [۳۰] بر روی شش سیگنال صوتی از پایگاه داده‌ی SQAM [۲۷] مقایسه کرده که نتایج آن در جدول ۲-۳ آمده است. نتایج این جدول نشان می‌دهد که برای اکثر سیگنال‌ها، به غیر از پیانو، نتایج فشرده‌سازی مبتنی بر تبدیل کمی بهتر هستند، اما بهبود میانگین نسبت به روش پیش بین خطی، کم است.

جدول ۲-۳. نتایج فشرده‌سازی روش LTAC در مقایسه با روش بدون اتلاف Shorten

	SHORTEEN	LTC (best fixed block length)	LTC (adaptive block length)
POP (Abba)	7.38	7.20	7.12
CLASSIC (Haydn)	5.95	5.90	5.87
PIANO (Schubert)	4.79	4.96	4.93
SOPRAN	6.22	5.90	5.88
MALE SPEECH	6.33	6.32	6.26
CASTANETS	7.51	7.04	6.90

بنابراین نتیجه‌ی این مقاله، این است که برای اکثر سیگنال‌ها، نرخ بیت‌ها که از کدگذاری آنروپی ضرایب و تصحیح خطا، حاصل شده‌اند، پایین‌تر از نتایج گرفته شده از پیش بین خطی است.

۲-۳-۳- روش مبتنی بر کدگذاری دوره‌ی تداوم

در مرجع [۲]، یک روش فشرده‌سازی بدون اتلاف که بسیار شبیه روش کدگذاری دوره‌ی تداوم^۱ می‌باشد، پیشنهاد شده است. در عمل، سیگنال صحبت شامل تعداد زیادی نمونه با مقدار صفر است. این، زمانی اتفاق می‌افتد که شخص در یک محیط با نویز کم صحبت می‌کند. در موسیقی، بعضی نمونه‌ها تغییرات تیز و سریع، و بعضی شامل صفرهای متوالی هستند. روش پیشنهادی، برای این قبیل سیگنال‌ها که تعداد صفرهای متوالی دارند، مناسب است. ممکن است در بعضی سیگنال‌ها اصلاً فشرده‌سازی انجام نشود.

روش پیشنهادی در این مقاله به این صورت است که اگر یک مقدار از سیگنال، به تعداد سه نمونه و یا بیشتر از سه مورد، به صورت پشت سر هم تکرار شود، این رشته‌ی تکراری نمادها را حذف کرده و آن‌ها را با سه نماد s_1 ، s_2 و s_3 جایگزین می‌کنیم. s_1 برابر مقدار آن نمونه‌ای است که تکرار شده است. s_2 مشخص کننده‌ی وجود تکرار در s_1 بوده و s_3 مشخص کننده‌ی تعداد تکرار s_1 است. مقداری که برای s_2 در نظر گرفته می‌شود، باید مقداری باشد که در بین نمونه‌های خود سیگنال موجود نباشد. در این مقاله پیشنهاد شده که اگر هم این مقدار در بین نمونه‌های خود سیگنال موجود باشد باید آن نمونه از سیگنال را اندکی تغییر دهیم. فرض کنید نمونه‌ها به صورت ۸ بیت PCM هستند. در این صورت مقادیر نمونه‌ها، مطابق نمایش متمم ۲-علامت دار، بین ۱۲۸- تا ۱۲۷+ خواهد بود. مقادیر ۱۲۰ و بالاتر از آن کمتر اتفاق می‌افتند. مقادیر نمونه‌ها، توزیع نرمال داشته و واریانس آن‌ها بسته به سیگنال، بین ۶ تا ۲۵ است [۳۱] (میانگین نمونه‌ها نزدیک صفر است). بنابراین نمونه‌ای با مقدار ۱۲۵ خیلی کم ظاهر می‌شود. مطابق رابطه‌ی توزیع نرمال، احتمال حضور

¹ Run-length Coding

مقدار ۱۲۵ با فرض $\sigma = 6$ برابر $3.7574 * 10^{-96}$ ، و با فرض $\sigma = 25$ برابر $5.9626 * 10^{-8}$ خواهد بود. مقدار ۱۲۷ خیلی کمیاب نیست زیرا تمامی مقادیر ۱۲۷ و بالاتر از آن، با مقدار ۱۲۷ ذخیره می‌شوند. در این جا، برای s_2 مقدار ۱۲۵ در نظر گرفته شده است. با استدلالی مشابه، اگر تعداد بیت‌های PCM، N بیت باشد، مقدار $s_2 = 2^N - 3$ پیشنهاد می‌شود. تمامی نمونه‌هایی از سیگنال را که مقدار آنها ۱۲۵ است، به ۱۲۶ تغییر می‌دهیم. در این صورت، اندکی تغییر در سیگنال خواهیم داشت. درصد تغییر انرژی، فقط برای این نمونه از سیگنال، $1 - \frac{125^2}{126^2} = 1.58\%$ خواهد بود. اگر نمادها را، ۸ بیت در نظر بگیریم، در حالتی که تعداد تکرارهای متوالی یک نمونه، بیشتر از ۲۵۵ باشد، دچار مشکل خواهیم شد. برای رفع این مشکل، به جای s_3 تعداد کلمات بیشتری قرار می‌دهیم. به عنوان مثال فرض کنید تعداد تکرار نماد صفر، ۵۴۰ مرتبه باشد. برای نمایش این تکرارها، ۵ نماد لازم خواهد بود. این مقادیر به صورت مقابل هستند: $s_1 = 0$ ، $s_2 = 125$ ، $s_3 = 255$ و $s_4 = 255$ و $s_5 = 29$. داریم: $255 + 255 + 29 = 539$. همانطور که می‌بینیم تعداد تکرار را تا جایی ادامه می‌دهیم که عدد مربوطه، کمتر از ۲۵۵ باشد.

در این روش اگر از تغییر کم ارزش‌ترین بیت، صرف نظر شود، نرخ فشرده‌سازی بهتر می‌شود. مثلاً در مورد یک سیگنال، با در نظر گرفتن تمام بیت‌ها، میزان کاهش طول سیگنال، ۲۱/۲۲٪ بوده، اما با صرف نظر از بیت کم ارزش، این مقدار به ۳۹/۴۷٪ رسیده است.

روش پیشنهادی با روش‌های فشرده‌سازی بدون اتلاف ^۱FLAC، Apple lossless، MPEG-4 ALS^۲، و Monkey's audio و Windows media 9 مقایسه شده است. برای ارزیابی، پنج صدای مرد و پنج صدای زن آزمایش شده‌اند. نرخ فشرده‌سازی در این روش، وابسته به سیگنال است. میزان کاهش طول سیگنال، بین ۱۰ الی ۲۸ درصد تغییر می‌کند. نتایج مقایسه برای این ده سیگنال، به طور میانگین در جدول ۲-۴ مشاهده می‌شود. معیار مقایسه در این جدول، نرخ فشرده‌سازی است و هرچه

^۱ Free Lossless Audio Codec

^۲ Audio Lossless Standard

این عدد بزرگ باشد، یعنی میزان فشردگی زیاد است. همان طور که از این جدول می‌بینیم، میزان فشردگی روش پیشنهادی در این مقاله در مقایسه با تمامی روش‌های دیگر پایین‌تر است. اما نسبت به همه‌ی آن روش‌ها ساده‌تر است. در این مقاله، مقایسه‌ی روش پیشنهادی با MPEG-4 ALS، کمی بیشتر بحث شده و با توجه به اینکه میزان فشردگی MPEG-4 حدود ۳۶/۴۰٪ بهتر از روش پیشنهادی است، گفته شده که مزیت روش پیشنهادی، سریع بودن آن و نیاز حجم سخت‌افزاری کمتر برای پیاده‌سازی سخت‌افزاری است. همچنین در مواردی که با روش پیشنهادی در این مقاله، فشردگی ممکن نباشد، سریع‌تر تذکر می‌دهد. همچنین گفته شده که به جای ارسال داده‌های تکراری متوالی در هر فریم، می‌توان داده‌های امنیتی را فرستاد که به این مورد، واترمارکینگ گفته می‌شود. در این صورت امنیت و شفافیت واترمارکینگ بیشتری خواهیم داشت [۲]. ما نیز در بخش فشردگی مبتنی بر تبدیل، برای کد کردن نمونه‌های تکراری متوالی، از روشی مشابه با این روش با کمی تغییر استفاده کرده‌ایم.

جدول ۲-۴. مقایسه نرخ فشردگی روش پیشنهادی در [۲] با پنج روش دیگر.

MPEG-4 ALS	FLAC	Apple lossless	Windows media 9	Monkey's audio	روش پیشنهادی در [2]
۱/۸۴۲	۱/۶۳۷	۱/۶۳۹	۱/۶۸۴	۱/۷۱۵	۱/۳۵

فصل سوم

فشرده‌سازی مبتنی بر تبدیل

در این فصل، ابتدا دو تبدیل DCT و DWT را که در این جا استفاده شده‌اند، معرفی می‌کنیم. سپس روش فشرده‌سازی مبتنی بر تبدیل را با تأکید بر تبدیل‌های DCT و DWT بیان خواهیم کرد. از این دو تبدیل برای مقایسه‌ی روش مبتنی بر وگت نیز استفاده شده است.

۳-۱- معرفی تبدیل موجک

تبدیل فوریه^۱ از زمان‌های پیش برای پردازش سیگنال‌های زمانی استفاده می‌شد. تبدیل فوریه یک تبدیل محلی^۲ نیست به این معنی که ضرایب تبدیل فوریه از روی تمام سیگنال در حوزه‌ی زمان محاسبه می‌شوند. و بنابراین سهم هر کدام از فرکانس‌ها در حوزه‌ی زمان مشخص نمی‌شود. در نتیجه اطلاعات حاصل از تبدیل فوریه مربوط به کل سیگنال بوده و هیچ اطلاعی از سهم حضور یک فرکانس خاص در یک زمان خاص نمی‌دهد. در نتیجه این تبدیل نمی‌تواند اطلاعات مربوط به سیگنال‌های غیر ایستاد^۳ را به خوبی آشکار کند. بنابراین این تبدیل برای استخراج اطلاعات فرکانسی از سیگنال‌های ایستاد^۴ مناسب است.

تبدیل فوریه‌ی کوتاه مدت یا STFT این مشکل را تا حدودی حل می‌کند. در این تبدیل، سیگنال در یک پنجره با طول محدود در زمان‌های مختلف ضرب شده و از آن تبدیل فوریه گرفته می‌شود. بنابراین تبدیل STFT یک تبدیل محلی به شمار می‌آید به طوری که مشخص می‌کند هر کدام از اطلاعات طیفی سیگنال، مربوط به کدام قسمت از سیگنال در حوزه‌ی زمان است. طول پنجره در تبدیل STFT، مشخص کننده‌ی رزولوشن زمانی و فرکانسی می‌باشد. مهم‌ترین عیب تبدیل STFT این است که رزولوشن زمانی و فرکانسی که با طول پنجره مشخص می‌شود برای تمام فرکانس‌ها یکسان است.

¹ Fourier Transform

² Local Transform

³ Nonstationary

⁴ Stationary

۳-۱-۱- تبدیل موجک

در مقابل تبدیل STFT که از طول پنجره‌های ثابت استفاده می‌کند، تبدیل موجک^۱ از یک روش پنجره گذاری با طول پنجره‌های متغیر بهره می‌گیرد [۳۲]. تبدیل موجک این اجازه را به ما می‌دهد که وقتی که اطلاعات دقیق‌تری از محتویات فرکانس پایین سیگنال را می‌خواهیم، از طول پنجره‌های بزرگ، و وقتی اطلاعات مربوط به محتویات فرکانسی بالای سیگنال مد نظر باشد، از طول پنجره‌های کوچک استفاده کنیم. یک راه برای رسیدن به این هدف، استفاده از توابع پایه‌ای با طول بزرگ و فرکانس پایین و هم‌چنین توابعی با طول کوچک و فرکانس بالا است. در تجزیه‌ی موجک، این توابع، با تغییر مقیاس دادن و شیفت دادن یک تابع اصلی $\psi(t)$ ، به دست می‌آیند. به چنین قابلیت‌ی قدرت زوم (یا تمرکز) کردن^۲ گفته می‌شود [۱۰].

تابع تحلیل در تبدیل موجک، تابع موجک نام دارد. تابع موجک، یک تابع موج گونه (نوسانی) با انرژی محدود است که مقدار آن سریعاً به صفر میل می‌کند یعنی بازه‌ی آن محدود است [۳۳]. ایده‌ی اصلی در تجزیه‌ی موجک، بیان یک سیگنال به صورت ترکیب خطی از یک مجموعه‌ی توابع، که از تغییر مقیاس یک تابع اصلی $\psi(t)$ به اندازه‌ی a ، و شیفت دادن آن به اندازه‌ی u ، $(\psi_{a,u}(t))$ ، به دست آمده‌اند می‌باشد. تابع $\psi(t)$ را موجک مادر یا تابع موجک می‌نامند.

$$\psi_{a,u}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-u}{a}\right) \quad (۱-۳)$$

که در آن $a, u \in \mathbb{R}$ و $a \neq 0$.

تابع $\psi(t)$ ، شکل موج‌های مختلفی می‌تواند داشته باشد. تبدیل موجک برخلاف تبدیل فوریه، به دنبال یافتن یک مولفه‌ی فرکانسی دوره‌ای^۳ خاص در سیگنال نیست. بلکه به دنبال یافتن جزئیات سیگنال به اندازه‌ی a ، در یک زمان مشخص است [۱۰]. از رابطه‌ی $\psi_{a,u}(t)$ معلوم است که با کوچک شدن a ، تابع پایه فشرده شده و در نتیجه فرکانس آن بالا می‌رود. بنابراین هنگام مقایسه‌ی موجک و

^۱ Wavelet Transform

^۲ Zooming Property

^۳ Circular Frequency

تبدیل فوریه، باید در نظر داشته باشیم که فرکانس و اندازه‌ی جزئیات a ، رابطه‌ی معکوس با هم دارند [۱۰].

تبدیل موجک پیوسته‌ی (CWT)^۱ یک سیگنال $s(t)$ ، به صورت (۲-۳) تعریف می‌شود:

$$C_{a,u} = \int_{-\infty}^{+\infty} s(t) \frac{1}{\sqrt{|a|}} \psi^* \left(\frac{t-u}{a} \right) dt = \langle s(t), \psi_{a,u}(t) \rangle \quad (2-3)$$

که در آن علامت * به معنی مزدوج مختلط و علامت $\langle \rangle$ نشان دهنده‌ی ضرب داخلی است.

در تبدیل موجک، سیگنال در زمان‌های مختلف در مقیاس‌های مختلف تحلیل می‌شود. یعنی در زمان‌های مختلف، اندازه‌ی جزئیات (فرکانس‌های) مختلف سیگنال حاصل می‌شوند. برای مقیاس‌های بزرگ، تابع پایه پهن‌تر بوده و ضرایب موجک متناظر، اطلاعات مربوط به محتوای فرکانسی پایین سیگنال را دارا می‌باشند. از طرف دیگر با کم کردن مقیاس a ، اطلاعات مربوط به محتوای فرکانسی بالای سیگنال به دست می‌آیند. بنابراین، رزولوشن زمانی زیاد در فرکانس‌های بزرگ، و رزولوشن فرکانسی بالا در پنجره‌های زمانی بزرگ (فرکانس‌های کوچک) حاصل می‌شود.

تجزیه‌ی موجک، منجر به تولید یک مجموعه از ضرایب به نام ضرایب موجک می‌شود. بنابراین سیگنال می‌تواند از ترکیب خطی از توابع موجک که با ضرایب موجک وزن دار شده‌اند بازسازی شود. برای بازسازی دقیق سیگنال باید تعداد کافی از ضرایب محاسبه شوند. تبدیل موجک پیوسته بر روی تمام صفحه‌ی \mathbb{R}^2 ، یعنی تمام زوج مرتب‌های حقیقی (a, u) ، تعریف می‌شود. اطلاعات موجود در CWT، در مقیاس‌ها و زمان‌های نزدیک به هم، شدیداً همبسته^۲ هستند یعنی با یکدیگر رابطه دارند. اصطلاحاً گفته می‌شود که ضرایب CWT، با یکدیگر تزیاید^۳ دارند. برای درک بهتر فرض کنید سیگنال نمونه‌برداری شده‌ی ما، N نمونه داشته باشد (با دوره‌ی تناوب T_s). اگر برای M مقیاس مختلف (a_1, a_2, \dots, a_M) ، ضرایب CWT را حساب کنیم داریم:

^۱ Continuous Wavelet Transform

^۲ Correlated

^۳ Redundancy

$$C_{a_i,KT_s} = (i = 1, \dots, M; K = 1, \dots, N) \quad (3-3)$$

یعنی در صفحه‌ی زمان-مقیاس، $M * N$ ضرایب خواهیم داشت. بنابراین تبدیل CWT هرچند که یک تحلیل محلی و چند مقیاسی از سیگنال ارائه می‌دهد، اما تعداد ضرایب آن بسیار زیاد بوده و این ضرایب شدیداً همبسته هستند. بنابراین این تبدیل برای فشرده‌سازی نمی‌تواند مفید باشد. علاوه از آن، تبدیل موجک پیوسته، محاسبات زیاد و سنگینی دارد. در تبدیل‌های مورد استفاده در فشرده‌سازی، لازم است که تعداد ضرایب تبدیل تقریباً با تعداد نمونه‌های سیگنال برابر باشد، و همچنین انرژی سیگنال تنها در تعداد کمی از این ضرایب متمرکز شده باشد، و البته سیگنال اصلی بتواند از روی ضرایب تبدیل بازسازی شود (یعنی تبدیل، برگشت پذیر باشد).

حال سؤال این است که آیا می‌توان تبدیلی پیدا کرد که مزایای تبدیل CWT را داشته باشد اما معایب آن را در مورد فشرده‌سازی نداشته باشد؟ به عبارت دیگر، آیا می‌توان تبدیلی پیدا کرد که مقادیر آن فقط روی یک زیرشبکه‌ی گسسته از صفحه‌ی زمان-مقیاس مربوط به CWT محاسبه شود، به طوری که تعداد ضرایب آن تقریباً برابر طول سیگنال اصلی باشد و سیگنال اصلی بتواند از روی این مقادیر به طور کامل بازسازی شود؟

جواب این سؤال مثبت است و در بخش بعدی (DWT) داده شده است.

۳-۱-۲ - تبدیل موجک گسسته یا DWT

تبدیل موجک گسسته^۱ با انتخاب ضرایب مقیاس و پارامترهای شیفیتی که توانی از ۲ هستند، حاصل می‌شود. یعنی $a = 2^j$ و $u = k2^j$ و $j, k \in \mathbb{Z}$ اگر این‌ها را در رابطه‌ی (۳-۱) قرار دهیم، داریم:

$$\psi_{j,k}(t) = 2^{-j/2} \psi(2^{-j}t - k) \quad (4-3)$$

تبدیل DWT، با جای گذاری رابطه‌ی بالا در رابطه‌ی (۳-۲)، به صورت زیر به دست می‌آید:

¹ Discrete Wavelet Transform

$$d_{j,k} = \int_{-\infty}^{+\infty} s(t)2^{-j/2}\psi^*(2^{-j}t - k)dt = \langle s(t), \psi_{j,k}(t) \rangle \quad (5-3)$$

که $d_{j,k}$ را، ضرایب موجک یا ضرایب جزئیات در مقیاس j و محل k ، می‌نامند. ضرایب $d_{j,k}$ در واقع مقادیر نمونه‌برداری شده از صفحه‌ی زمان-مقیاس تبدیل موجک پیوسته هستند. اما یکی از مزایای اصلی این ضرایب این است که بین آن‌ها اطلاعات زائد و تکراری وجود ندارد. حال با استفاده از DWT می‌توان سیگنال اصلی را به طور کامل و دقیق بازسازی کرد. به علاوه، تبدیل DWT می‌تواند به صورت یک تحلیل چند مقیاسی¹ تفسیر شود. طوری که یک سلسله مراتب از تقریب‌ها و جزئیات سیگنال، حاصل می‌شوند.

تجزیه‌ی چند مقیاسی سیگنال $s(t)$ ، در سطح تجزیه‌ی H ، به صورت زیر تعریف می‌شود:

$$\begin{aligned} s(t) &= \sum_{k=-\infty}^{+\infty} a_{H,k}2^{-H/2}\varphi(2^{-H}t - k) + \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} d_{j,k}2^{-j/2}\psi(2^{-j}t - k) \\ &= A_H(t) + \sum_{j=-\infty}^{+\infty} D_j(t) \end{aligned} \quad (6-3)$$

$\psi(t)$ ، همان تابع موجک و $\varphi(t)$ ، یک تابع همراه است که تابع مقیاس² نام دارد. توابع $\varphi_{j,k}(t) = 2^{-H/2}\varphi(2^{-H}t - k)$ ، نسخه‌های تغییر مقیاس داده شده و شیفت یافته‌ی تابع مقیاس اصلی، $\varphi(t)$ ، هستند. $a_{H,k}$ معرف ضرایب تقریب³ (یا ضرایب مقیاس⁴) در سطح تجزیه‌ی H است که از کانال⁵ کردن سیگنال، با تابع مقیاس $\varphi_{H,k}(t)$ به دست آمده است ($a_{H,k} = \langle s(t), \varphi_{H,k}(t) \rangle$). با نگاه به رابطه‌ی (6-3)، می‌بینیم که سیگنال به دو بخش تقریب ($A_H(t)$) در سطح تجزیه‌ی H ، که قسمت اعظم انرژی سیگنال را دارد، و یک توالی از سیگنال‌های جزئیات (از سطح H ، تا منفی بی‌نهایت) تجزیه شده است. سیگنال جزئیات $D_j(t)$ ، در واقع سیگنال خطای باقی مانده‌ای است که

¹ Multi Resolution

² Scale Function

³ Approximation Coefficient

⁴ Scale Coefficient

⁵ Convolve

وقتی از سیگنال تقریب در وضوح^۱ بالاتر (مقیاس کمتر $j - 1$) به سیگنال تقریب در وضوح پایین تر (مقیاس بزرگتر j) می‌رویم، ایجاد می‌شود.

با ادامه دادن تجزیه، روی تمام سطوح، بسط موجک به صورت زیر حاصل می‌شود:

$$s(t) = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} d_{j,k} 2^{-j/2} \psi(2^{-j}t - k) \quad (7-3)$$

رابطه‌ی (۷-۳)، بازسازی سیگنال اصلی $s(t)$ را از روی ضرایب موجک، $d_{j,k}$ ، بیان می‌کند. از لحاظ تئوری، تجزیه‌ی موجک، تمامی مقیاس‌ها را از $-\infty$ تا $+\infty$ ، شامل می‌شود (رابطه‌ی (۷-۳)). در عمل تجزیه‌ی موجک، روی سیگنال‌های گسسته در زمان (با دوره‌ی تناوب نمونه‌برداری T_s) اعمال می‌شود. مقادیر اندازه گرفته شده برای نمونه‌های سیگنال، عموماً مربوط به ضرایب تقریب در سطح $j = 0$ می‌باشند و بنابراین تجزیه‌ی موجک، محدود به مقیاس‌های مثبت می‌شود [۳۴] (بین دو نمونه‌ی متوالی از سیگنال اصلی، وضوح به اندازه‌ی کافی وجود ندارد تا سطوح تجزیه به مقیاس $j < 0$ برسد). از طرف دیگر طول سیگنال محدود است، و بنابراین مقادیر مربوط به k نیز محدود هستند. با توجه به مطالب گفته شده و همچنین توجه به رابطه‌ی $(d_{j,k})$ ، در می‌یابیم ارتباط بین ضرایب CWT و DWT به صورت رابطه‌ی زیر است:

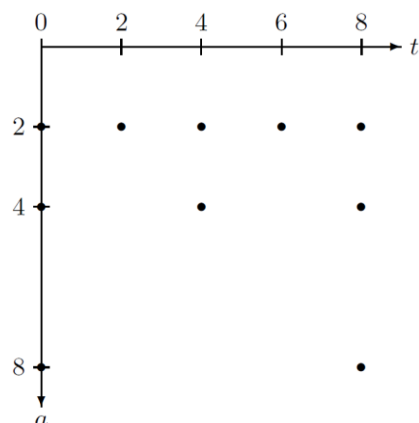
$$C_{2^j, 2^j k} = d_{j,k} \quad (j = 1, \dots, J) \quad (8-3)$$

بنابراین محل ضرایب DWT در صفحه‌ی زمان-مقیاس به صورت شکل ۱-۳ خواهد بود.

مطابق با الگوریتم مالات^۲ [۳۵]، تبدیل موجک گسسته که با رابطه‌ی (۵-۳) بیان شد، می‌تواند به صورت یک فیلتربانک با فیلترهای پایین‌گذر و بالاگذر یکسان که به صورت سری قرار می‌گیرند پیاده شود (شکل ۲-۳ الف)). در شکل ۲-۳ الف)، $h[n]$ فیلتر پایین‌گذر و $g[n]$ فیلتر

¹ Resolution

² Mallat

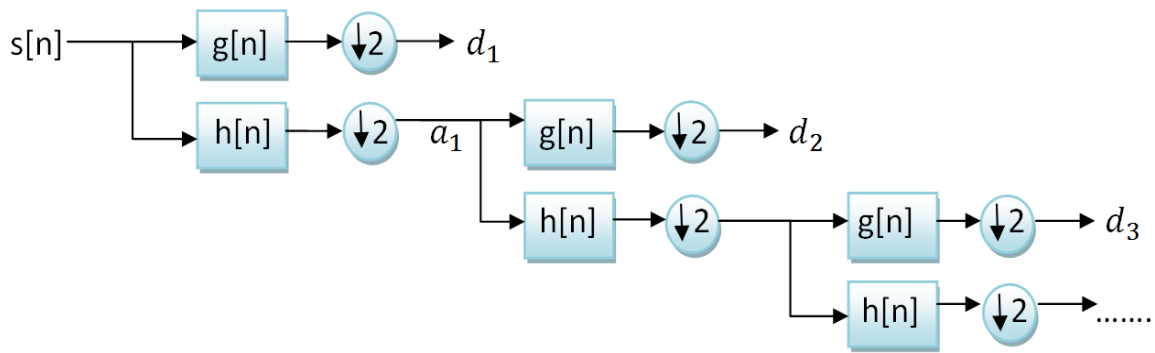


شکل ۳-۱. نقاط مربوط به DWT در صفحه‌ی زمان-مقیاس

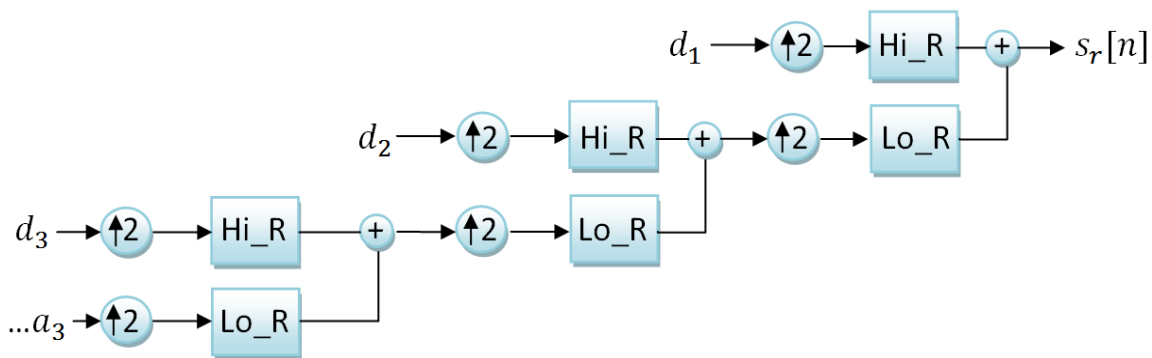
بالاگذر است. فیلترهای بالاگذر با توابع موجک $\psi_{j,k}(t)$ و فیلترهای پایین‌گذر با توابع مقیاس $\phi_{j,k}(t)$ مرتبط هستند. در هر مرحله، این زوج فیلتر، سیگنال را به دو بخش ضرایب تقریب (خروجی فیلتر پایین‌گذر) و ضرایب جزئیات (خروجی فیلتر بالاگذر) تجزیه می‌کنند. عمل فیلتر کردن ابتدا به سیگنال اصلی اعمال می‌شوند. و در مراحل بعدی، همین فیلترها به قسمت تقریب به دست آمده از مرحله‌ی قبل اعمال می‌شوند. مطابق شکل در هر مرحله، بر روی خروجی هر کدام از فیلترها زیرنمونه‌برداری انجام می‌شود. این زیرنمونه‌برداری، باعث حذف تزیاید موجود در بین ضرایب می‌شود و همچنین با نصف کردن طول سیگنال خروجی، حجم محاسبات را برای تجزیه‌های بیشتر کاهش می‌دهد.

عکس عمل زیرنمونه‌برداری، در مرحله‌ی بازسازی سیگنال با یک بالانمونه‌بردار انجام شده است که باعث می‌شود طول سیگنال بازسازی شده با سیگنال اصلی یکی شود. در مرحله‌ی بازسازی هم، از دو فیلتر پایین‌گذر و بالاگذر دیگر که هر کدام با بالانمونه‌برداری همراه هستند، استفاده می‌شود (شکل ۳-۲(ب)). فیلترهای پایین‌گذر و بالاگذر مربوط به تجزیه، به همراه دو فیلتر پایین‌گذر و بالاگذر مربوط به بازسازی، یک سیستم به نام فیلترهای آینه‌ی تریبوعی^۱ [۳۶] را تشکیل می‌دهند.

^۱ Quadrature Mirror Filters (QMF)



(الف)



(ب)

شکل ۲-۳. الف) فیلتربانک مربوط به تجزیه‌ی DWT. ب) فیلتربانک مربوط به بازسازی

بنابراین با توجه به مطالب گفته شده نتیجه می‌گیریم که تجزیه‌ی J-مرحله‌ای DWT یک الگوریتم ساده و سریع برای محاسبه‌ی تبدیل موجک است که حداقل سه حُسن دارد: اول اینکه تعداد ضرایب به دست آمده تقریباً با طول سیگنال برابر است؛ دوم اینکه ضرایب به دست آمده از تبدیل موجک گسسته، با یکدیگر تزیاد ندارند؛ و سوم اینکه برای محاسبه‌ی مقادیر تبدیل موجک، از زیر مجموعه‌ای از نقاط صفحه‌ی زمان-مقیاس استفاده شده است که شامل ضرایب مقیاس متفاوت می‌باشد. یعنی این یک تجزیه‌ی چند مقیاسی است. علاوه از این‌ها، DWT یک الگوریتم سریع می‌باشد که شامل فیلتر کردن ساده است.

همچنین نکات زیر هم شایان توجه هستند:

تبدیل موجک قابلیت تجزیه‌ی چند مقیاسی را دارد به طوری که دقت محل یابی^۱ زمانی و فرکانسی، خود به خود با محتوای فرکانسی قسمت مورد نظر از سیگنال وفق داده می‌شود. از این رو تبدیل موجک مخصوصاً برای آنالیز ناپیوستگی‌ها، تغییرات سریع، قسمت‌های غیر متناوب و دیگر ویژگی‌های غیر ایستان سیگنال، مناسب است به طوری که تغییرات موجود در سیگنال را در مقیاس‌های مناسب آشکار می‌کند. از این رو از تبدیل موجک، هم در فشرده‌سازی سیگنال‌های صوتی [۳۸،۳۷] و هم سیگنال‌های صحبت [۳۶] استفاده می‌شود.

به علاوه، بر خلاف تبدیل فوریه که فقط توابع سینوسی به عنوان توابع پایه‌ی تبدیل استفاده می‌شوند، تعداد متنوعی از شکل موج‌ها میتوانند به عنوان موجک مادر استفاده شوند. و این یکی دیگر از مزایای اصلی تبدیل موجک است که اجازه می‌دهد تا مناسب‌ترین تابع پایه متناسب با یک کاربرد خاص، انتخاب شود.

یک تفاوت مهم بین تبدیل موجک و سایر تبدیلات سینوسی این است که توابع پایه‌ی منفرد در تبدیل موجک در حوزه‌ی زمان، محلی هستند یعنی هر کدام در قسمت کوچکی از زمان دارای مقدار بوده و برای بقیه‌ی زمان‌ها صفر هستند. اما در تبدیلات سینوسی، توابع پایه، که توابع سینوسی و کسینوسی با فرکانس‌ها، فازها و دامنه‌های مختلف هستند، محلی نبوده و برای تمام زمان‌ها مقدار دارند. تأثیر این تفاوت را بعداً در سیگنال‌های بازسازی شده در قسمت نتایج خواهیم دید.

۳-۲- تبدیل DCT

در حالت کلی، تبدیلات، روشی برای بیان یک سیگنال برحسب یک عده توابع پایه هستند. توابع پایه در تبدیل فوریه، نمایی‌های مختلط هستند که از توابع کسینوسی به عنوان بخش حقیقی و توابع سینوسی به عنوان بخش موهومی تشکیل شده‌اند. در تبدیل فوریه گسسته، هم تبدیل‌های پایه

¹ Localization

و هم ضرایب تبدیل، در حالت کلی مختلط هستند. یک سیگنال می‌تواند به صورت کامل، فقط بر حسب توابع پایه‌ی کسینوسی و یا سینوسی بیان شود. این امر منجر به تولید یک دسته از تبدیلات به نام تبدیل‌های کسینوسی گسسته (DCT) و تبدیل‌های سینوسی گسسته (DST) می‌شود [۳۹]. در تبدیل DCT، هم دنباله‌های پایه و هم ضرایب آن‌ها حقیقی هستند. از تبدیل DCT به صورت خیلی گسترده در فشرده‌سازی تصویر و صوت و استخراج ویژگی استفاده شده است [۳۹].

تبدیل DCT و معکوس آن به وسیله‌ی رابطه‌ی (۹-۳) محاسبه می‌شوند:

$$X[k] = 2 \sum_{n=0}^{N-1} a[n]x[n] \cos\left(\frac{\pi kn}{N-1}\right) \quad 0 \leq k < N-1$$

$$x[n] = \frac{1}{N-1} \sum_{k=0}^{N-1} a[k]X[k] \cos\left(\frac{\pi kn}{N-1}\right) \quad 0 \leq n \leq N-1 \quad (9-3)$$

که در آن، $a[n]$ به صورت رابطه‌ی (۱۰-۳) تعریف می‌شود:

$$a[n] = \begin{cases} \frac{1}{2} & n = 0, N-1 \\ 1 & 1 \leq n \leq N-2 \end{cases} \quad (10-3)$$

یکی ویژگی مهم DCT، توانایی آن در متمرکز کردن بیشتر انرژی سیگنال در تعداد کمی از ضرایب است. بعضی از اولین روش‌های فشرده‌سازی سیگنال‌های صوتی با کیفیت بالا، بر پایه‌ی تبدیل فوریه گسسته و تبدیل کسینوسی گسسته بودند.

۳-۳- روش فشرده‌سازی مبتنی بر تبدیل

تبدیلات، ابزاری هستند که یک تابع یا دنباله را به یک تابع یا دنباله‌ی دیگر نگاشت می‌کنند. بعضی از مزایای استفاده از تبدیل‌ها به جای توابع اصلی، این است که کارکردن با تبدیل، معمولاً راحت‌تر از توابع اصلی است. تبدیل‌ها ممکن است حافظه‌ی کمتری برای ذخیره‌سازی احتیاج داشته باشند که این مسأله باعث فشرده‌سازی می‌شود و ممکن است انجام عملیات ریاضی بر روی تابع تبدیل آسانتر از تابع اصلی باشد [۴۰].

انگیزه‌ی اصلی در استفاده از هر تبدیل، بیان یک تابع پیچیده (در این جا، سیگنال) با استفاده از توابع ساده است [۴۱]. مجموعه‌ی توابعی که برای نمایش یک تابع به کار می‌روند، توابع پایه نامیده می‌شوند. ابتدا یک تابع به بخش‌های کوچکتر خود شکسته شده و هر کدام از این بخش‌ها با نسخه‌های شیفت یافته و تغییر مقیاس شده‌ی تابع پایه، نمایش داده می‌شوند. هر کدام از تبدیل‌ها توابع پایه‌ی متفاوتی دارند. به عنوان مثال توابع پایه در تبدیل DCT، کسینوسی‌ها و توابع پایه در تبدیل DWT، بستگی به تابع موجک دارد. با توجه به عملیات پایه‌ای که در تبدیلات انجام می‌شود، آن‌ها می‌توانند به طور مؤثر برای فشرده‌سازی سیگنال‌های صوتی استفاده شوند.

روش‌های فشرده‌سازی مبتنی بر تبدیل، از مزایای موجود در سیگنال‌های صوتی بهره می‌گیرند تا نمایش‌های مؤثرتری در حوزه‌ی تبدیل ایجاد کنند. در واقع هر چه مزایای در بین ضرایب تبدیل کمتر باشد، مناسب‌تر خواهد بود. دلیل اصلی در کاهش نرخ بیت با روش‌های مبتنی بر تبدیل، این است که این تبدیلات، محتوای تبدیلی غیر همبسته^۱ با هم تولید می‌کنند [۴۱]. کارایی یک سیستم فشرده‌سازی مبتنی بر تبدیل، به نوع آن تبدیل خطی و نحوه‌ی کد کردن ضرایب بستگی خواهد داشت [۴۲]. فشرده‌سازی در روش‌های مبتنی بر تبدیل، با تمرکز کردن انرژی سیگنال در تعداد کمی از ضرایب تبدیل انجام می‌شود. هر چقدر که انرژی بیشتر در تعداد ضرایب کمتر متمرکز شود، خطای ایجاد شده از کوانتیزاسیون کمتر خواهد بود [۴۲].

تبدیل‌های مختلفی برای فشرده‌سازی استفاده شده‌اند به عنوان مثال می‌توان به تبدیل کسینوسی گسسته، تبدیل فوریه‌ی گسسته، تبدیل WHT^۲، تبدیل موجک گسسته و غیره اشاره کرد.

همچنین بعضی روش‌ها، از ترکیب تبدیل‌های مختلف استفاده کرده‌اند. توابع پایه‌ی دو یا چند تبدیل، که معمولاً متعامد نیستند، برای تبدیل‌های ترکیبی استفاده می‌شوند [۴۳]. این روش‌ها سعی می‌کنند تا با استفاده از چند شکل موج نمونه که مطابقت بهتری با ویژگی‌های محلی سیگنال صوتی

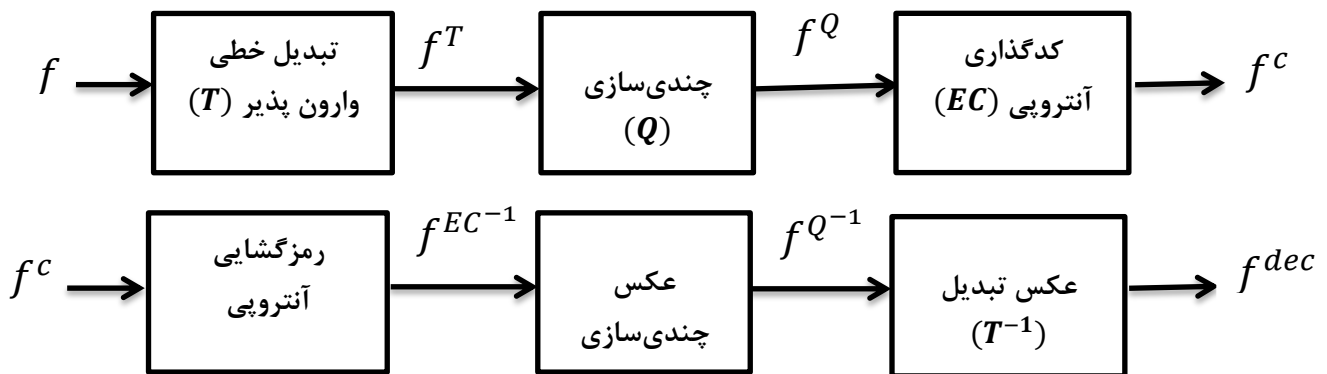
^۱ Uncorrelated

^۲ Walsh-Hadamard Transform

دارند، یک نمایش دقیقتری از سیگنال فراهم کنند. در [۴۴] از ترکیب تبدیل‌های فوریه و WHT استفاده شده است. در [۴۵] از ترکیب DWT و DCT برای فشرده‌سازی سیگنال‌های صوتی استفاده شده است.

۳-۳-۱- فرآیند کلی طرح فشرده‌سازی مبتنی بر تبدیل

طرح فشرده‌سازی مبتنی بر تبدیل از سه مرحله‌ی کلی تشکیل شده است. این فرآیند در شکل ۳-۳ نشان داده شده است.



شکل ۳-۳. نمودار بلوکی الگوریتم کلی فشرده‌سازهای مبتنی بر تبدیل

بلوک دیاگرام فوق حالت کلی بوده و جزئیات بیشتر در متن روش‌ها توضیح داده خواهد شد. هر کدام از این مراحل مستقل از هم نیستند و باید به گونه‌ای انتخاب شوند که در نهایت، نرخ بیت پایین و کیفیت بالا داشته باشیم. در مرحله‌ی اول، یک تبدیل خطی برگشت پذیر بر روی سیگنال ورودی اعمال می‌شود. در مرجع [۴۶]، سه تبدیل FFT، DCT و DWT برای سیگنال‌های صحبت با هم مقایسه شده‌اند. نتایج آن‌ها نشان می‌دهد که عملکرد DCT و DWT نسبت به FFT با اختلاف خیلی زیاد بهتر است. بنابراین ما هم در این پایان نامه از دو تبدیل DCT و DWT استفاده کرده‌ایم که جزئیات هر کدام در جای خود توضیح داده خواهد شد.

در مرحله‌ی بعد، آستانه‌گذاری و کوانتیزاسیون انجام می‌شود. کوانتیزاسیون یکی از مراحل مهم در فشرده‌سازی می‌باشد که باید در انتخاب نوع کوانتیزاسیون، مرحله‌ی سوم از بلوک دیاگرام شکل ۳-۳، یعنی کدگذاری آنتروپی را هم در نظر بگیریم. کوانتیزاسیون به دو شکل عددی و برداری می‌باشد. کوانتیزاسیون برداری زمانی مفید است که بین داده‌ها، همبستگی وجود داشته باشد. اما با اعمال تبدیل، محتویات تبدیل دیگر همبستگی ندارند و اصلاً این یکی از دلایل اصلی استفاده از تبدیل در فشرده‌سازی است و بنابراین برای ضرایب تبدیل میتوان از کوانتیزاسیون عددی استفاده کرد [۴۱]. هم‌چنین روش‌های کوانتیزاسیون با حافظه‌ای مثل DPCM و ADPCM که از همبستگی موجود در سیگنال برای فشرده‌سازی بیشتر استفاده می‌کنند، این جا نمی‌توانند مفید باشند. لذا ما هم در این قسمت از چندی‌سازی یکنواخت استفاده کرده‌ایم.

آخرین گام در فشرده‌سازی مبتنی بر تبدیل همان طور که در شکل ۳-۳ نشان داده شده است، کدگذاری آنتروپی است. مشابه مرحله‌ی اول (یعنی تبدیل)، این مرحله نیز برگشت‌پذیر است و در نتیجه مشکلات مربوط به تقریب را که در قسمت کوانتیزاسیون داشتیم، این جا نداریم.

در ادامه، دو روش مبتنی بر تبدیل با استفاده از تبدیل موجک و DCT را که دو مورد از مؤثرترین تبدیلات در فشرده‌سازی صوت و صحبت هستند توضیح می‌دهیم. در این پایان نامه، یک بررسی کلی از این دو روش انجام گرفته و سعی شده که بهترین نتایج حاصل شود. همچنین از این دو روش برای مقایسه‌ی روش جدید پیشنهادی استفاده شده است. شایان ذکر است که با توجه به ملاحظات انجام شده، نتایج حاصل از هر دو روش در این جا از تمامی مقادیر آمده در مراجع قبلی که از این روش‌ها استفاده کرده‌اند، بهتر است.

۳-۴- روش فشرده‌سازی مبتنی بر تبدیل DWT

در طراحی یک سیستم فشرده‌سازی مبتنی بر تبدیل موجک، چندین پارامتر نقش مهمی دارند.

۳-۴-۱- انتخاب تابع موجک بهینه

انتخاب تابع موجک در طراحی سیستم‌های فشرده‌سازی صوت با کیفیت بالا، از اهمیت بالایی برخوردار است. توابع موجک، از خانواده‌ی موجک‌های مختلفی هستند که هر کدام می‌توانند برای یک کاربرد خاص مناسب باشند. معیارهای مختلفی می‌تواند در تعیین تابع موجک بهینه نقش داشته باشد. هدف، مینم کردن خطای سیگنال بازسازی شده و رسیدن به سیگنال به نویز بالا است.

یک معیار مناسب برای انتخاب موجک‌های بهینه، میزان انرژی است که تابع پایه‌ی موجک می‌تواند در ضرایب تقریب مربوط به سطح تجزیه‌ی ۱ متمرکز کند [۴۷]. هم‌چنین انتخاب تابع موجکی که تعداد ممان‌های صفر شونده‌ی^۱ به اندازه‌ی کافی زیاد داشته باشد، برای فشرده‌سازی صحبت ضروری است [۴۸]. با توجه به این که معیارهای دیگری نیز از قبیل آنتروپی ضرایب و ... در نتیجه‌ی فشرده‌سازی تأثیر گذار هستند، یک راه مناسب، بررسی موجک‌های مختلف و انتخاب آن از روی نتایج تجربی است. بررسی کارهای گذشته نشان می‌دهد که موجک db10، نتایج بهتری نسبت به db6، db8 و هم‌چنین سایر توابع، از خانواده‌های دیگر دارد [۴۶، ۴۷]. در [۴۹] گفته شده که db15، شفافیت ادراکی صحبت را نسبت به توابع موجک دیگر بیشتر حفظ می‌کند. با توجه به این که db15، تعداد ممان‌های صفر شونده‌ی بیشتری هم دارد، بنابراین ما هم این موجک را برای آزمایش استفاده کرده‌ایم.

۳-۴-۲- تعداد سطوح تجزیه

یکی از مواردی که در فشرده‌سازی مبتنی بر تبدیل DWT باید در نظر گرفت، انتخاب سطح تجزیه‌ی مناسب برای تبدیل است. انتخاب سطح تجزیه برای DWT، معمولاً به نوع سیگنال و بعضی معیارهای دیگر مثل آنتروپی بستگی دارد. برای سیگنال‌های صحبت ادامه‌ی تجزیه تا سطح ۵ مناسب است و بعد از آن، دیگر بهبودی در عملکرد سیستم دیده نمی‌شود [۵۰]. در [۴۷] هم نشان داده شده

¹ Vanishing Moments

که برای db10 بعد از سه سطح تجزیه، تغییر قابل توجهی چه برای صدای زن و چه صدای مرد، حاصل نمی‌شود. نتایج تجربی نشان می‌دهد که در سطوح تجزیه‌ی بالا، عملکرد سیستم ضعیف‌تر هم می‌شود [۵۱]. ما هم در این جا آزمایش‌ها را با پنج سطح تجزیه انجام داده‌ایم.

۳-۴-۳ - آستانه‌گذاری برای صفر کردن ضرایب

بعد از اعمال تبدیل موجک بر روی سیگنال، اکثر ضرایب تبدیل به دست آمده صفر بوده و یا مقدار ناچیزی دارند. یعنی اطلاعات سیگنال در تعداد کمی از ضرایب حفظ می‌شود. به طوری که با حذف بقیه‌ی ضرایب، اتلاف بسیار کمی خواهیم داشت. در [۵۰] نشان داده شده که موجک Battle-Lemarie، بیش از ۹۷/۵٪ از انرژی سیگنال را در قسمت تقریب از ضرایب، متمرکز می‌کند. توابع موجک دیگر هم به همین شکل هستند و بنابراین با صفر کردن ضرایب کوچک می‌توان به نرخ فشرده‌سازی بالا دست یافت.

دو روش برای این کار وجود دارد. روش آستانه‌گذاری سراسری^۱، و روش آستانه‌گذاری وابسته به سطح^۲ که از استراتژی بیرگ-ماسارت^۳ استفاده می‌کند [۱۴]. در روش آستانه‌گذاری سراسری، در بین ضرایب تبدیل DWT، درصد مشخصی از این ضرایب را که کوچک هستند، صفر می‌کنند. اما در آستانه‌گذاری وابسته به سطح، برای هر سطح تجزیه، تعداد مشخصی از ضرایب را که متناظر آن سطح هستند، باقی گذاشته و مابقی را حذف می‌کنند. نتایج آزمایش‌ها نشان می‌دهد که آستانه‌گذاری سراسری، نسبت به آستانه‌گذاری وابسته به سطح، عملکرد بهتری دارد [۵۲، ۴۷]. به همین دلیل، ما نیز از آستانه‌گذاری سراسری استفاده کرده‌ایم.

¹ Global Thresholding

² Level-Dependent Thresholding

³ Birge Massart

۳-۴-۴- کوانتیزاسیون

کوانتیزاسیون یکی از بخش‌های مهم فشرده‌سازی به شمار می‌رود. در کوانتیزاسیون تعداد بسیار زیاد و پراکنده‌ای از داده‌های ورودی به تعداد محدودی نماد نگاشت می‌شوند که این مسأله باعث می‌شود تا آنتروپی داده‌ها به شدت کاهش یابد و علاوه از آن، هر کدام از نمادها می‌توانند با تعداد بیت‌های کمتری نمایش داده شوند. کوانتیزاسیون یک مرحله‌ی با اتلاف است. بنابراین در این مرحله، باید روشی را انتخاب کنیم که تلفات کم و نرخ فشرده‌سازی بیشتر داشته باشد. در فصل ۲، سه روش کوانتیزاسیون معرفی کردیم. همانطور که گفته شد، با توجه به این که ضرایب تبدیل با یکدیگر همبستگی ندارند، کوانتیزاسیون با حافظه، گزینه‌ی خوبی نیست. هم‌چنین روش $\mu - law$ ، برای سیگنال‌های حوزه‌ی زمان است چرا که با کوانتیزه کردن دقیق‌تر مقادیری که دامنه‌ی کم دارند، باعث افزایش سیگنال به نویز می‌شود. بنابراین ما از روش کوانتیزاسیون یکنواخت استفاده کرده‌ایم.

۳-۴-۵- نحوه‌ی کدگذاری نمادهای تکراری

بعد از آستانه‌گذاری و کوانتیزاسیون، تعداد زیادی ضرایب صفر خواهیم داشت. لازم است که یک روش مناسب برای کد کردن ضرایب تبدیل استفاده کنیم. یک راه برای این کار این است که ضرایب غیرصفر را به همراه محل آن‌ها در بردار تبدیل، ذخیره کنیم [۵۲]. در این صورت برای هر ضریب غیرصفر، دو بایت خواهیم داشت. یکی محل آن ضریب و دیگری مقدار آن. راه دیگر آن است که صفرهای متوالی را با دو بایت نشان دهیم [۴۷]، مشابه آنچه که در کدگذاری دوره‌ی تداوم گفته شد. نتایج پیاده‌سازی ما نشان می‌دهد که روش دوره‌ی تداوم بهتر از روش اولی، که ضرایب را با محل آن‌ها ذخیره می‌کند، جواب می‌دهد. دلیل آن، این است که وقتی که ضرایب را با محل آن‌ها ذخیره می‌کنیم به ازای هر ضریب غیرصفر، یک بایت خواهیم داشت که نماینده‌ی محل آن ضریب است و مقدار آن متفاوت از همه‌ی بایت‌های دیگر است (زیرا محل هر ضریب متفاوت است). بنابراین آنتروپی بسیار بالایی خواهد داشت. اما در روش دوره‌ی تداوم، بایستی که برای مشخص کردن شروع تکرار صفر

به کار می‌رود، همه جا یکسان است یعنی در کل، آنتروپی کمی داشته و بنابراین وقتی که از کدگذاری آنتروپی استفاده کنیم، نرخ فشرده‌سازی، هرچه بیشتر خواهد بود. مشاهدات ما نشان می‌دهد که علاوه از صفر، نمادهای دیگر هم تکرارهای متوالی دارند. بنابراین ما در اینجا از روشی مشابه با روش ارائه شده در [۲] با کمی تغییر استفاده کرده‌ایم. بدین صورت که برای تعیین بایت مشخص کننده‌ی شروع تکرار از روشی متفاوت بهره می‌گیریم. به این صورت که هنگام کوانتیزاسیون، اگر نمایش نمادها قرار است با B بیت انجام شود، سطوح کوانتیزاسیون را $2^B - 1$ در نظر می‌گیریم و نه 2^B . در این صورت نمادهای ما می‌توانند اعداد 0 تا $2^B - 2$ باشند. لذا نماد با مقدار $2^B - 1$ را می‌توان به عنوان مشخص کننده‌ی شروع تکرار استفاده کرد. در این صورت اعوجاجی که در مقادیر سیگنال در [۲] ایجاد می‌شد را نخواهیم داشت.

۳-۴-۶- کد گذاری آنتروپی

آخرین مرحله در فشرده‌سازی مبتنی بر تبدیل، کدگذاری آنتروپی است. بعد از کوانتیزاسیون و کدگذاری نمادهای تکراری، کدگذاری آنتروپی می‌تواند مؤثر باشد. باید توجه نمود که در کدگذاری آنتروپی، اگر طول فریم‌ها کوتاه باشد، فشرده‌سازی مؤثری انجام نمی‌شود و در بعضی مواقع، سیگنال کدگذاری شده، طولانی‌تر از سیگنال قبل از کدگذاری خواهد بود. بنابراین لازم است تا هنگام فریم بندی سیگنال اصلی، فریم‌ها را تا اندازه‌ی کافی بزرگ در نظر بگیریم.

کد گذاری آنتروپی را در فصل ۲ توضیح دادیم. با توجه به این که کدگذاری حسابی نسبت به هافمن بهینه‌تر است [۱۰]، ما هم از کدگذاری حسابی استفاده کردیم. در این روش بعد از این که کدگذاری انجام شد و رشته‌ی بیت خروجی (کد یا رمز خروجی) تولید شد، این رمز به تنهایی برای رمزگشایی کافی نخواهد بود. اولاً با توجه به توضیحات داده شده در فصل ۲ باید طول رشته‌ی نمادهای ورودی را هم بفرستیم. ثانیاً باید احتمالات حضور هر نماد یا تعداد تکرار هر نماد در رشته‌ی ورودی را هم ارسال کنیم.

برای این که بتوان از این روش به صورت پیش رونده^۱ استفاده کرد و همچنین برای استفاده در کاربردهای زمان حقیقی، لازم است که سیگنال ورودی، فریم‌بندی شود. با توجه به این که در انتها از کدگذاری آنتروپی استفاده می‌شود لذا همان طور که گفته شد، هر چه که طول فریم‌ها بزرگتر باشد، تأثیرگذاری آنتروپی بیشتر خواهد بود.

۳-۵- روش فشرده‌سازی مبتنی بر تبدیل DCT

این روش مانند روش DWT است با این تفاوت که به جای DWT از تبدیل DCT استفاده شده است، و بنابراین مسائلی را که در مورد انتخاب نوع موجک و سطح تجزیه و انواع آستانه‌گذاری سراسری و وابسته به سطح مطرح بود، این جا نداریم. بنابراین روش DCT نسبت به DWT بسیار ساده‌تر است.

در این روش ابتدا سیگنال ورودی فریم‌بندی می‌شود، سپس به هر فریم، تبدیل DCT اعمال می‌شود. در این صورت ضرایب DCT حاصل می‌شوند. تعداد این ضرایب با تعداد نمونه‌های فریم ورودی یکسان است. تبدیل DCT نیز مشابه DWT این خاصیت را دارد که بیشتر اطلاعات سیگنال را تنها در تعداد کمی از ضرایب تبدیل، متمرکز می‌کند و اکثر ضرایب حاصل، مقدار کمی دارند. بنابراین با صفر کردن این ضرایب و نگه داشتن ضرایب با دامنه‌ی بزرگ، اتلاف کمی در سیگنال حاصل می‌شود. از این خاصیت به منظور فشرده‌سازی استفاده می‌شود. بنابراین با آستانه گذاری، ضرایب با دامنه‌ی پایین را صفر می‌کنیم. بعد از این مرحله نوبت به کوانتیزاسیون می‌رسد. با توجه به نبود همبستگی در ضرایب تبدیل، این جا هم به صورت یکنواخت کوانتیزه می‌کنیم. بعد از آن کدگذاری حسابی را اعمال می‌کنیم.

یک دلیل استفاده‌ی زیاد از تبدیل DCT در فشرده‌سازی صوت این است که سیگنال‌های صوتی ماهیت شبه متناوب دارند و بنابراین تبدیلات سینوسی برای تحلیل آن‌ها مناسب هستند. در

^۱ Progressive

[۵۳] آمده است که در فشرده‌سازی صحبت، عملکرد DCT نزدیک به بهینه است زیرا بردارهای DCT، بردارهای ویژه‌ی یک گروه بزرگی از ماتریس‌های توپلیتز^۱ هستند.

^۱ Toeplitz

فصل چہارم

وجلتها

با توجه به این که روش پیشنهادی در این پایان نامه، مشابه وجلت‌ها بوده و در واقع از وجلت‌ها گرفته شده است، لذا در این فصل، تئوری وجلت‌ها را که یک ابزار برای پردازش تصاویر هستند، به طور خلاصه معرفی می‌کنیم تا آشنایی کافی حاصل شود. اما برای تعمیم این روش به سیگنال‌های صوتی، باید تغییراتی انجام دهیم که توضیحات آن در قسمت مربوطه خواهد آمد.

وجلت‌ها^۱ برای اولین بار توسط دونوهو [۵۴] در سال ۱۹۹۷ مطرح شدند و ارتباط نزدیکی با موجک‌ها دارند. دونوهو، سال‌های زیادی را در زمینه‌ی موجک‌ها تحقیق و پژوهش کرده و مقالات زیادی را به چاپ رسانده است. مهم‌ترین کاربرد وجلت‌ها، فشرده‌سازی تصویر، حذف نویز و ناحیه بندی تصویر و تشخیص لبه است. رامبرگ^۲ و همکارانش از وجلت‌ها در فشرده‌سازی چندمقیاسی تصاویر استفاده کرده‌اند [۵۵، ۵۶]. همان‌طور که در [۵۶] نشان داده شده است، فشرده‌ساز مبتنی بر وجلت، در فشرده‌سازی تصویر، عملکرد بهتری حتی نسبت به استاندارد JPEG2000 دارد.

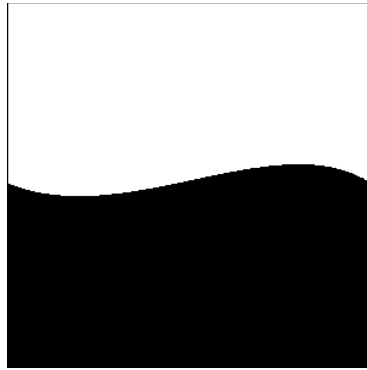
۴-۱- وجلت

ایده‌ی اصلی وجلت، مدل کردن تغییرات بافت تصویر است که این کار با جدا کردن دو بافت از هم با استفاده از لبه‌هایی صورت می‌گیرد. وجلت‌ها برای بار اول در مورد تصاویر ساده و باینری ارائه شدند اما بعدها برای همه‌ی تصاویر و با کاربردهای مختلف مورد استفاده قرار گرفتند که کاربردهای آن روز به روز در حال افزایش است. برای درک بهتر این ابزار، بحث را با یک مثال ساده بیان می‌کنیم. شکل ۴-۱ یک تصویر باینری است که لبه‌هایی بین قسمت سفید و سیاه آن وجود دارد. مسأله این است که چطور بتوانیم این لبه را با یک روش منطقی مدل بکنیم. یک راه ساده این است که همانند شکل ۴-۲(الف) یک خط راست از محل تقاطع لبه با قسمت راست تصویر تا محل تقاطع آن به قسمت چپ تصویر رسم کنیم. اما اگر لبه‌ی مورد نظر انحنای زیادی داشته باشد یک خط راست

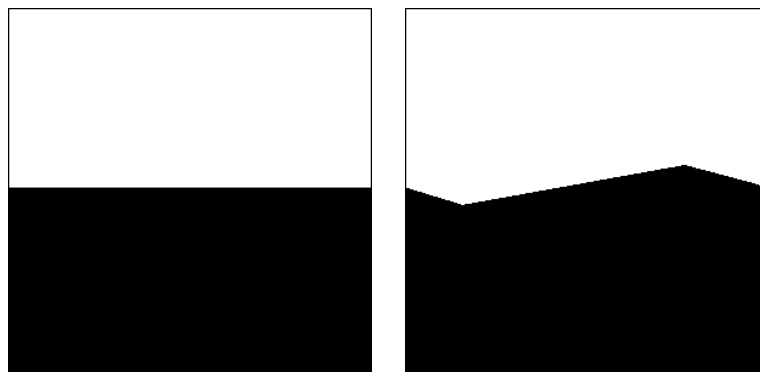
¹ Wedgelet

² Romberg

نمی‌تواند تقریب خوبی باشد. در این صورت باید از چند خط به هم متصل همانند شکل ۴-۲(ب) استفاده کرد. مسأله این است که چطور بتوانیم نقاط شکستگی لبه را در شکل ۴-۲(ب) پیدا بکنیم. وجلت‌ها این مسأله را حل کرده‌اند.



شکل ۴-۱. یک تصویر باینری با لبه‌هایی که تصویر را به دو بخش سیاه و سفید تقسیم کرده است



(الف)

(ب)

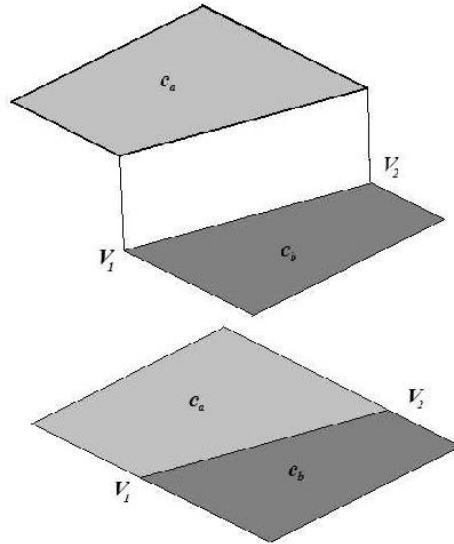
شکل ۴-۲. (الف) تقریب با یک خط. (ب) تقریب با تکه‌های خطی

تعریف وجلت: یک وجلت، w ، یک تابع بر روی مربع d است که مربع را با یک لبه‌ی e به دو قسمت بخش می‌کند که هر قسمت سطح روشنایی ثابتی با مقادیر c_a و c_b دارند. محل تلاقی e را با مربع d ، v_1 و v_2 می‌نامیم.

بنابراین وجلت را می‌توان به صورت زیر نشان داد.

$$w = \{c_a, c_b, e\} = \{c_a, c_b, v_1, v_2\} \quad (1-4)$$

شکل ۳-۴ تصویری از یک وجلت را نشان می‌دهد. برای تعیین مقادیر c_a و c_b ، میانگین پیکسل‌هایی را که بالای لبه e قرار می‌گیرند، c_a ، و میانگین پیکسل‌هایی را که زیر لبه e قرار می‌گیرند c_b می‌نامیم.



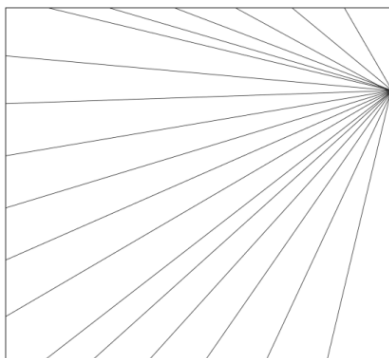
شکل ۳-۴. یک مثال از یک وجلت

تعریف: یک وجلت هم‌تراز^۱، یک تابع بر روی مربع d است که این تابع در کل d ، یک مقدار ثابت c دارد.

می‌توان این گونه فرض کرد که وجلت هم‌تراز، یک وجلت است که لبه e در آن از داخل مربع d عبور نمی‌کند. در این صورت ما دو نوع وجلت خواهیم داشت. وجلت هم‌تراز و وجلت غیر هم‌تراز.

در داخل یک مربع خاص از یک تصویر، برای پیدا کردن لبه‌ی مورد نظر، تمام حالت‌های مختلف لبه‌ها (وجلت) را برای آن مربع امتحان می‌کنیم و هر کدام به تصویر نزدیک‌تر باشد آن را انتخاب می‌کنیم. شکل ۴-۴ تمام لبه‌های ممکن در یک مربع را که از یک نقطه رد می‌شوند نشان می‌دهد.

¹ Degenerate Wedgelet



شکل ۴-۴. تمام حالات پیکربندی ممکن از یک رأس واقع بر محیط یک مربع در یک مقیاس خاص

معیار تعیین نزدیک‌ترین لبه به تصویر در بین لبه‌های یک مربع، مقدار CPRSS^۱ است:

$$CPRSS = \|y - \mu\|^2 + \lambda \cdot \#P \quad (۲-۴)$$

در این رابطه y ، مقدار اصلی پیکسل‌های تصویر، و μ ، مقدار پیکسل‌ها بعد از تجزیه‌ی وجلت است. λ یک ثابت و $\#P$ ، مربوط به پیچیدگی درخت تجزیه می‌باشد. در واقع معیار CPRSS، تابع هزینه می‌باشد که شامل دو جمله می‌باشد. جمله‌ی اول مربوط به مجموع مربع خطا بوده و جمله‌ی دوم مربوط به پیچیدگی درخت تجزیه است. ثابت λ ، بین پیچیدگی و خطا، موازنه ایجاد می‌کند. هر لبه که مقدار CPRSS برای آن کم باشد، بهینه‌تر خواهد بود.

می‌توان نشان داد که موازنه‌ی پیچیدگی بهینه در تجزیه‌ی وجلت به ازای λ ی زیر به دست می‌آید:

$$\lambda = (\zeta \cdot \sigma(1 + \sqrt{2 \log_e(\#W)}))^2 \quad (۳-۴)$$

که در آن $\zeta < 8$ یک ثابت است و $\#W$ ، تعداد تمام وجلت‌های موجود در دیکشنری است که در بخش دیکشنری وجلت این کمیت را با M نشان داده‌ایم. برای تشریح کامل و بحث بیشتر در مورد این رابطه، می‌توانید به [۵۴] مراجعه کنید.

۴-۱-۱- تجزیه‌ی وجلت‌ها

ابتدا فرض می‌کنیم تصویر I ، به صورت $[0,1] \times [0,1]$ نرمالیزه شده است.

^۱ Complexity Penalized Residual Sum of Squares

فرض می‌کنیم مربع $S(k_1, k_2, j)$ را داریم:

$$S(k_1, k_2, j) = [k_1/2^j, (k_1 + 1)/2^j] \times [k_2/2^j, (k_2 + 1)/2^j] \quad (4-4)$$

که در آن $0 \leq k_1, k_2 < 2^j$ ، برای اعداد صحیح $j \geq 0$.

به عبارت دیگر فرض می‌کنیم که تصویر I را به صورت $N \times N$ جدول بندی کنیم

که $N = 2^j$. حال هر کدام از مربع‌های حاصل را با سه عدد j ، k_1 و k_2 ، میتوانیم با توجه به رابطه‌ی

(4-4) به راحتی مشخص کنیم. اگر j را افزایش دهیم، مربع‌ها کوچک‌تر خواهند شد. مشخص است

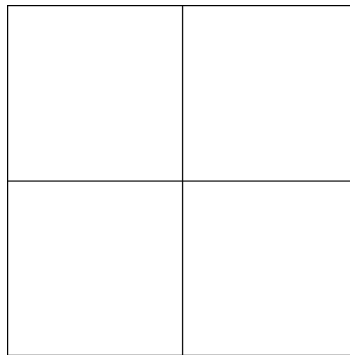
که $S(0,0,0)$ بیان‌کننده‌ی کل تصویر می‌باشد.

اگر $j = 0$ باشد یعنی یک وجلت داریم و آن کل تصویر است. اگر $j = 1$ باشد یعنی چهار

وجلت داریم به نحوی که کل تصویر به 4 قسمت تقسیم شده است همانند شکل 4-5. اگر $j = 2$

باشد یعنی هر کدام از مربع‌های (وجلت‌های) شکل 4-5، خود به چهار مربع دیگر تقسیم می‌شوند و

این کار ادامه می‌یابد.

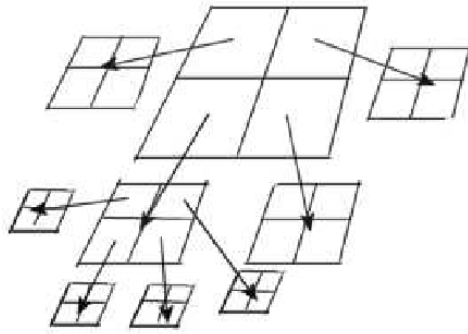


شکل 4-5. یک مرحله انشعاب در تجزیه‌ی وجلت

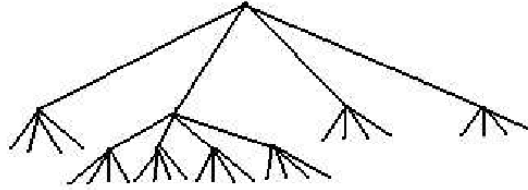
شکل 4-6 تصویر خوبی از تجزیه‌ی وجلت‌ها را نشان می‌دهد. به این تجزیه درخت چهارگانه¹ هم گفته

می‌شود.

¹ Quadtree



(الف)



(ب)

شکل ۴-۶. الف) شکل یک تجزیه‌ی وِجِلت. هر کدام از وِجِلت‌ها می‌توانند هم تراز یا غیر هم تراز باشند. ب) درخت چهارگانه مربوط به تجزیه‌ی قسمت الف

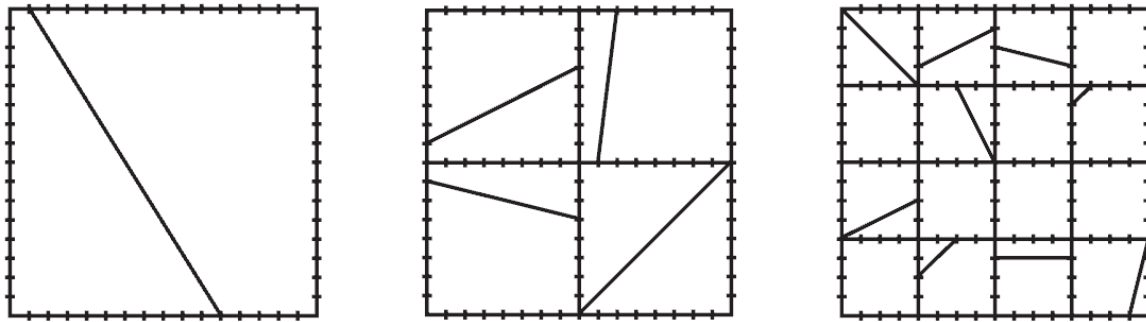
۴-۱-۲- دیکشنری وِجِلت

وقتی که می‌خواهیم یک تصویر را با وِجِلت‌ها مدل کنیم، یک حداکثر سطح تجزیه را انتخاب می‌کنیم و آن را با J نشان می‌دهیم. هر چه J بزرگ باشد مدل ما به تصویر اصلی نزدیک‌تر خواهد بود. فرض می‌کنیم تصویری روی $[0,1] \times [0,1]$ داریم که آن را به صورت $N \times N$ جدول بندی کرده‌ایم (یا به عبارت دقیق‌تر، به مربع‌های کوچکی با ضلع $1/N$). روی این تصویر وِجِلت‌هایی با اندازه‌های مختلف می‌توانیم داشته باشیم. می‌توان این‌گونه فرض کرد که بر روی اضلاع هر کدام از مربع‌های $S(k_1, k_2, j)$ و $0 \leq k_1, k_2 < 2^j$ ، رأس‌هایی^۱ با فاصله‌ی $1/N$ وجود دارند. طبیعتاً هر چه اندازه‌ی وِجِلت بزرگ باشد (مربع بزرگی باشد) تعداد رأس‌های روی اضلاع آن وِجِلت بیشتر می‌شود.

با وصل کردن هر دو تا از این رأس‌ها به صورت یک خط، یک لبه به وجود می‌آید. به این لبه، بیملت^۲ [۵۷] هم می‌گویند. این بیملت‌ها (لبه‌ها) می‌توانند در وِجِلت‌هایی با اندازه‌های مختلف رسم شوند. شکل ۴-۷ یک تصویر خوبی از رأس‌ها و بعضی لبه‌ها نشان می‌دهد.

^۱ Vertices

^۲ Beamlet



شکل ۴-۷. افراز تصویر در سطوح مختلف و رأس‌ها و بعضی لبه‌ها

تعداد رأس‌های موجود بر روی یک مربع $S(k_1, k_2, j)$ و $0 \leq k_1, k_2 < 2^j$ برابر
 $M(S(k_1, k_2, j)) = 4 * 2^{-j} * N$ است که به راحتی قابل فهم است. بنابر این تعداد کل لبه‌هایی
 که در این وجلت می‌توانیم داشته باشیم برابر $\binom{M(S(k_1, k_2, j))}{2}$ است.

حال تمام مربع‌های ممکن موجود را در نظر بگیرید. یعنی $S(k_1, k_2, j)$ ، $0 \leq k_1, k_2 < 2^j$

و $0 \leq j < J$. تعداد کل لبه(بیملت)های ممکن موجود در کل تصویر، M ، برابر است با [۵۴]:

$$M = \sum_{j=0}^J \sum_{k_1, k_2=0}^{2^j-1} \binom{M(S(k_1, k_2, j))}{2} \approx 8 * (\log_2(N) + 1) * N^2 \quad (۵-۴)$$

برای اثبات این رابطه، می‌توانید به [۵۴] مراجعه کنید. پس تا این جا تمام لبه‌های ممکن موجود در
 وجلت‌های یک تصویر را شمردیم. این لبه‌ها همگی با هم دیکشنری وجلت را تشکیل می‌دهند.

به راحتی مشخص است که این لبه‌ها کل لبه(بیملت)های ممکن موجود در تصویر را پوشش
 نمی‌دهند چرا که ممکن است از دو رأس موجود در دو وجلت مختلف به هم وصل کنیم که بر روی
 لبه‌های موجود در دیکشنری منطبق نباشد. تعداد کل لبه‌های ممکن تصویر برابر N^4 است که نسبت
 به M خیلی بزرگ است اما این دیکشنری همان طور که نشان خواهیم داد قادر است تمام تصویر را با
 دقت خوبی نمایندگی کند. دلیل آن به این حقیقت بر می‌گردد که دیکشنری شامل لبه‌هایی در
 جاهای زیاد و با جهت‌ها و اندازه‌های مختلف است.

۴-۱-۳ - الگوریتم تجزیه‌ی و جلت (درخت تجزیه‌ی بهینه)

الگوریتم تجزیه‌ی و جلت می‌تواند از پایین به بالا و یا از بالا به پایین باشد. الگوریتمی که در پایین توضیح داده می‌شود از پایین به بالا می‌باشد. الگوریتم به این صورت است که ابتدا تصویر را با ماکزیمم درجه‌ی تفکیک، J ، تفکیک می‌کنیم. یعنی در ابتدا j را برابر با J قرار می‌دهیم. بدین معنی که از کوچک‌ترین مربع‌ها شروع می‌کنیم.

- درخت تجزیه‌ی کامل را تا سطح $\log_2 N$ ، بر روی تصویر بساز.
- برای هر گره از درخت تجزیه مقادیر زیر را حساب کن:
 - $CPRSS \leftarrow a_s$ مربوط به و جلت هم تراز (بدون لبه).
 - $\min(CPRSS) \leftarrow b_s$ مربوط به و جلت غیر هم تراز.
 - لبه‌ای که b_s برای آن حاصل شد $e_s =$
- $j = J = \log_2 N$
- حلقه: برای هر مربع S ، در سطح تجزیه‌ی j ،
 - اگر $j < \log_2 N$ ، قرار بده $d_s = \sum_i c_{s_i}$ (مجموع تمام زیر مربع‌های S). اگر
 - $j = \log_2 N$ ، قرار بده $d_s = +\infty$ (هیچ زیرمربعی وجود ندارد)
 - $c_s = \min(a_s, b_s, d_s)$
- اگر $c_s = a_s$ و یا $c_s = b_s$ ، و جلت S را به همراه e_s ، نگه داشته و تمامی زیر مربع‌های آن را (در صورت وجود) از درخت تجزیه حذف کن. (e_s را فقط در صورتی ذخیره می‌کنیم که $c_s = b_s$)
- اگر $c_s = d_s$ ، همان زیر مربع‌های قبلی را به عنوان گره نهایی فعلی در نظر بگیر.

• $j = j - 1$

• اگر $j \geq 0$ به ابتدای حلقه برو.

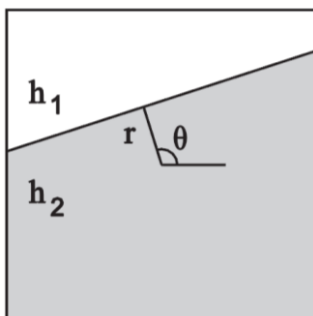
۴-۲- وجلت‌های توسعه یافته^۱

وجلت‌های با لبه‌ی خطی در مورد فشرده‌سازی تصاویر نتایج خیلی رضایت بخشی ندارند [۵۸]. در تصاویر، لبه‌ها، اغلب از خطوط راست تشکیل نشده‌اند بلکه بیشتر به صورت منحنی هستند (نه لزوماً منحنی‌های صاف). استفاده از وجلت‌های با لبه‌ی خطی، برای تقریب این منحنی‌ها مستلزم این است که تعداد خیلی زیادی از این مربع‌ها استفاده کنیم که در کاربردهای فشرده‌سازی به صرفه نیست. بنابراین لازم است تقریب‌هایی که در درون مربعات انجام می‌شود به صورت منحنی باشد. البته این را هم باید در نظر گرفت که تعمیم وجلت‌ها به وجلت‌هایی با لبه‌های منحنی، نباید پارامترهای زیادی را وارد دیکشنری وجلت کند. در [۵۸] نشان داده شده است که اضافه کردن یک پارامتر دیگر به دیکشنری، نتایج رضایت بخشی را به همراه دارد و باعث بهبود عملکرد سیستم فشرده‌ساز می‌شود. قبل از این که به توضیح دیکشنری وجلت‌های توسعه یافته برویم، یک نوع دیگر نمایش وجلت‌های اولیه را معرفی می‌کنیم.

۴-۲-۱- نمایش دیکشنری وجلت با پارامترهای θ و r

وجلت‌های معرفی شده را می‌توانیم به صورتی دیگر هم نشان دهیم. در این روش لبه را با دو پارامتر r (فاصله‌ی مرکز مربع تا مرکز لبه) و θ (زاویه‌ی خط افقی با خط مربوط به فاصله‌ی r) طبق شکل ۴-۸ مشخص می‌کنیم [۵۶]. علاوه بر آن مقادیر میانگین پیکسل‌های بالای لبه را h_1 و پایین لبه را h_2 می‌نامیم. با این مدل ما می‌توانیم تصاویر خاکستری را هم به آسانی آنالیز کنیم. در تصاویر باینری می‌توان در نظر گرفت $h_1 = h_2 = 1$.

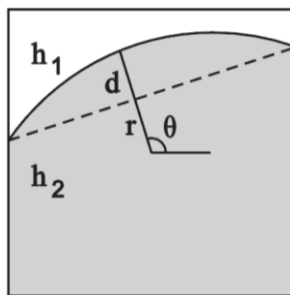
¹ Extended Wedgelets



شکل ۴-۸. نوع دیگر پارامتر گذاری و جلت ها

۴-۲-۲- دیکشنری و جلت های توسعه یافته

در بعضی تصاویر و جلت های با لبه ی به صورت خط راست جواب خوبی نمی دهد و یا این که باید سطح تجزیه را بسیار زیاد کنیم که حجم حافظه و محاسبات زیاد می شود. در این موارد و جلت هایی ارائه می شوند که لبه ی آن ها به صورت خط صاف نیست و منحنی است. این منحنی می تواند قسمتی از دایره یا سهمی یا هر چیز دیگر باشد. اما معمولاً سهمی جواب بهتری می دهد. با اضافه کردن یک پارامتر d به r و θ ، این کار را انجام می دهیم. با توجه به شکل ۴-۹، اگر خط مشخص کننده θ را ادامه دهیم، d ، فاصله ی تقاطع این خط با منحنی و لبه ی فرضی قدیمی است.



شکل ۴-۹. پارامتر گذاری و جلت های توسعه یافته.

ایده ی و جلت های توسعه یافته در [۵۸] مطرح شده و هر دو روش و جلت های اولیه (WD) و و جلت های تعمیم یافته (EWD) برای فشرده سازی تعدادی تصویر کارتونی آزمایش شده اند. نتایج این آزمایش را در جدول ۴-۱ می توانید ببینید. در این آزمایش، کیفیت تصاویر را با دو معیار $PSNR^1$ و

¹ Peak Signal to Noise Ratio (PSNR)

CDM¹ [۵۹] اندازه گرفته‌اند. همان طور که از این جدول می‌بینیم، در کیفیت تقریباً برابر، تعداد وجلت‌های استفاده شده در روش وجلت‌های توسعه یافته، حدود ۲۸٪ کمتر از تعداد وجلت‌های روش WD است.

جدول ۴-۱. نتایج فشردگی تصاویر کارتونی با دو روش تجزیه‌ی وجلت‌ها (WD) و وجلت‌های توسعه یافته (EWD) [۵۸]

	PSNR (dB)		CDM ₀ (%)		Number of wedgelets		$\frac{m_w}{m_w} \cdot 100\%$ (%)
	WD	EWD	WD	EWD	WD	EWD	
chicken	34,928	34,721	98,36	98,26	5848	4396	75,17
caution	31,938	32,214	98,22	98,27	6844	4840	70,72
figures	24,670	24,760	82,62	81,55	1426	943	66,13
shape	35,865	35,873	95,53	95,33	904	706	78,10
isobars	19,210	19,437	58,07	59,01	181	118	65,19

¹ Closest Distance Metric

فصل پنجم

روش پیشنهادی

در فصل قبلی، وجلت‌های اصلی را معرفی کردیم. در این فصل می‌خواهیم تعمیم این ابزار فشرده‌سازی سیگنال را روی سیگنال‌های صوتی بررسی کنیم. برای این کار توابع پایه‌ی وجلت‌ها را توابع چندجمله‌ای از مراتب مختلف انتخاب کرده‌ایم. بنابراین در ابتدای این فصل، مرور کوتاه و مختصری بر تئوری تقریب چندجمله‌ای خواهیم داشت.

۵-۱- چند جمله‌ای‌ها

چندجمله‌ای‌ها در ریاضیات و مخصوصاً در آنالیز عددی، بحث مهمی به شمار می‌آیند. توابع غیرخطی مختلف، می‌توانند بر روی یک بازه‌ی مشخص، با هر دقت مطلوبی تقریب زده شوند اگر ما از چندجمله‌ای‌هایی با مرتبه‌ی به اندازه‌ی کافی بزرگ استفاده کنیم [۶۰]. اگر چه حل معادلات چندجمله‌ای، بعضی وقت‌ها سخت می‌باشد، اما چندجمله‌ای‌ها در تعیین مقدار یک تابع مشخص، بسیار مفید هستند. در این گونه موارد، مشکل مربوط به حل معادلات چندجمله‌ای را نداریم. بنابراین، در تئوری تقریب، چندجمله‌ای‌ها، نسبت به توابع دیگر، نقش و اهمیت برتری دارند [۶۱].

یک تابع $P(x)$ به صورت رابطه‌ی (۵-۱) را یک چندجمله‌ای از x می‌نامند.

$$P(x) = a_0 + a_1x + a_2x^2 + \dots + a_mx^m \quad (۵-۱)$$

مقادیر a_0, a_1, \dots, a_m ، ضرایب چندجمله‌ای هستند و می‌توانند اعداد حقیقی یا مختلط باشند. بالاترین توان x را در یک چندجمله‌ای، مرتبه‌ی^۱ آن چندجمله‌ای تعریف می‌کنند. اگر مرتبه‌ی چندجمله‌ای m باشد، به تعداد $m + 1$ ضریب خواهد داشت.

۵-۱-۱- تقریب تابع

قبل از ورود به بحث تئوری تقریب، لازم است بگوییم که دو نوع مسأله می‌تواند مطرح باشد. یک نوع مسأله، زمانی است که تابع، به صورت کامل و واضح در دست باشد، اما ما بخواهیم که آن را با

^۱ Order

یک تابع ساده‌تر مانند چندجمله‌ای بیان کنیم به طوری که این تابع جدید، تقریبی از مقادیر تابع داده شده، به ما بدهد. نوع دوم مسأله زمانی است که ورودی ما، یک مجموعه از داده باشد و ما بخواهیم بهترین تابعی را پیدا کنیم که می‌تواند برای نمایش این مجموعه‌ی داده استفاده شود (حالت گسسته). با توجه به اینکه سیگنال‌های صوتی، به صورت گسسته در زمان هستند، ما با نوع دوم مسأله سروکار داریم. توابع مختلفی می‌توانند برای تقریب استفاده شوند، اما برای فشرده‌سازی، تقریب با استفاده از توابع چندجمله‌ای، مناسب‌تر است [۶۰]. در ادامه، روش تقریب مینیمم مربع خطا را که برای تقریب توابع چندجمله‌ای استفاده می‌شود توضیح می‌دهیم.

۵-۱-۲- تقریب مینیمم مربع خطای گسسته

فرض می‌کنیم مجموعه‌ی نقاط $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ را داشته باشیم. ایده‌ی اصلی در تقریب به روش مینیمم مربع خطا، این است که یک تابع چندجمله‌ای $P(x)$ از مرتبه‌ی m ($m + 1 \leq n$) را بر روی مجموعه‌ی نقاط داده شده پیدا کنیم، با این هدف که مجموع مربعات خطاها، بر روی این مجموعه نقاط گسسته، مینیمم شود. رابطه‌ی مربع خطا که باید مینیمم شود، به صورت زیر نوشته می‌شود:

$$S = \sum_{i=1}^n [y_i - P(x_i)]^2 \quad (2-5)$$

که در آن $P(x)$ به صورت رابطه‌ی (۱-۵) است. با جاگذاری رابطه‌ی (۱-۵) در (۲-۵)، داریم:

$$S = \sum_{i=1}^n [y_i - (a_0 + a_1x + a_2x^2 + \dots + a_mx^m)]^2 \quad (3-5)$$

برای مینیمم کردن S ، باید مشتق جزئی S نسبت به هر کدام از a_i ها را برابر صفر قرار دهیم. با انجام این کار به معادلات زیر می‌رسیم:

$$\frac{\partial S}{\partial a_0} = -2 \sum_{i=1}^n [y_i - (a_0 + a_1x + a_2x^2 + \dots + a_mx^m)] = 0$$

$$\frac{\partial S}{\partial a_1} = -2 \sum_{i=1}^n [y_i - (a_0 + a_1x + a_2x^2 + \dots + a_mx^m)].x_i = 0$$

$$\frac{\partial S}{\partial a_2} = -2 \sum_{i=1}^n [y_i - (a_0 + a_1x + a_2x^2 + \dots + a_mx^m)].x_i^2 = 0$$

.....

(۴-۵)

$$\frac{\partial S}{\partial a_m} = -2 \sum_{i=1}^n [y_i - (a_0 + a_1x + a_2x^2 + \dots + a_mx^m)].x_i^m = 0$$

این معادلات، پس از ساده سازی، به صورت کلی زیر نوشته می شوند:

$$s_0a_0 + s_1a_1 + \dots + s_ma_m = t_0$$

$$s_1a_0 + s_2a_1 + \dots + s_{m+1}a_m = t_1$$

.....

(۵-۵)

$$s_ma_0 + s_{m+1}a_1 + \dots + s_{2m}a_m = t_m$$

که در آن S و t ، به صورت (۶-۵) تعریف شده اند:

$$s_k = \sum_{i=1}^n x_i^k \quad ; \quad t_k = \sum_{i=1}^n y_i x_i^k \quad (۶-۵)$$

این سیستم را به نام سیستم معادلات نرمال می شناسند. با حل دستگاه معادلات (۵-۵) به

راحتی، ضرایب a_0, a_1, \dots, a_m به دست می آیند. اگر این ضرایب را در رابطه ی (۱-۵) قرار دهیم، به

بهترین تابع چندجمله ای از مرتبه ی m خواهیم رسید که می تواند داده های ورودی را نمایش دهد.

۵-۲- روش فشرده‌سازی مبتنی بر وجلت

روش پیشنهادی از نوع فشرده‌سازهای شکل‌موج^۱ می‌باشد زیرا فشرده‌سازهای شکل‌موج بیشترین کیفیت سیگنال را تأمین می‌کنند [۶۲]. فشرده‌سازهای شکل‌موج بدون در نظر گرفتن ویژگی‌های ادراکی صوت، سعی در کد کردن دقیق منحنی اصلی شکل‌موج سیگنال صوتی را دارند [۶۲]. با توجه به تفاوت‌های موجود بین سیگنال‌های صوت و تصویر، برای تعمیم روش وجلت در تصویر بر روی سیگنال یک بعدی صوتی نیاز به تغییراتی در الگوریتم این روش، نحوه‌ی تقریب، تعیین پارامترهای تابع هزینه و... داریم. در ادامه روش پیشنهادی را توضیح می‌دهیم.

۵-۲-۱- فریم بندی

اگر بخواهیم روش ارائه شده به صورت پیش رونده باشد، و یا این که بتواند در کاربردهای بلادرنگ استفاده شود، باید فریم بندی انجام دهیم. یعنی سیگنال را ابتدا به فریم‌هایی بخش کنیم و درخت تجزیه را در داخل آن فریم تشکیل داده و پس از طی مراحل گفته شده و انجام الگوریتم، کدگذاری آنروپی را اعمال کنیم. این فریم بندی را همان طور که در روند نمای شکل ۵-۳ می‌بینید، در ابتدا انجام می‌دهیم. این فریم بندی همچنین باعث می‌شود تا حجم حافظه‌ی لازم برای اجرای برنامه کم شود. با توجه به این که در این الگوریتم از کدگذاری آنروپی استفاده می‌شود، بنابراین هر چه طول فریم بیشتر باشد، جواب بهتری خواهیم داشت. از طرفی نباید طول فریم‌ها خیلی زیاد باشد که نتواند برای همه سیگنال‌ها استفاده شود و یا تأخیر زیادی داشته باشد. در اینجا، ما طول فریم‌ها را حدود ۸۱۹۲ نمونه در نظر گرفته‌ایم.

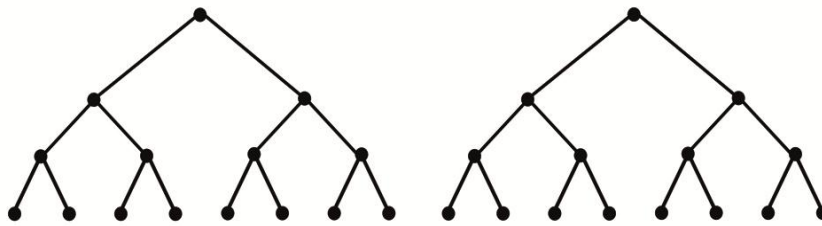
۵-۲-۲- درخت تجزیه

با توجه به این که طول سیگنال‌های صوتی، برای تقریب زدن بسیار طولانی هستند بنابراین نرمالیزه کردن طول کل سیگنال به بازه‌ی $[0,1]$ ، ایده‌ی خوبی نیست. علاوه از آن، همیشه کل

¹ Waveform Coders

سیگنال ورودی، از ابتدا تا انتها در دست نیست و همچنین در کاربردهای بلادرنگ و پیش رونده نمی توان کل سیگنال ورودی را یک جا تحلیل کرد.

ابتدا سیگنال صوتی را به فریم‌هایی (پنجره‌هایی) با طول 2^J افراز می‌کنیم. با توجه به این که طول این پنجره‌ها ثابت بوده و همچنین توانی از ۲ است، بنابراین برخلاف وجلت تصویر، نیازی به نرمالیزه کردن به بازه‌ی $[0,1]$ را نداریم. یک دیدگاه به این مسأله، این است که سیگنال را در پنجره‌ی به طول 2^J به روشی که خواهیم گفت، تقریب بزنیم. حال اگر با این تقریب، دقت لازم حاصل نشد، پنجره را به دو پنجره‌ی کوچک‌تر با طول‌های 2^{J-1} افراز می‌کنیم. دوباره سیگنال را در هر کدام از این دو پنجره، تقریب می‌زنیم. این افراز پنجره و تقریب را تا جایی ادامه می‌دهیم که به دقت مطلوب برسیم و یا اینکه دیگر جایی برای افراز وجود نداشته باشد. یکی از مزایای این روش این است که سیگنال را در محل‌های مختلف و در مقیاس‌های مختلف (همانند موجک‌ها) تقریب می‌زنیم. آخرین سطح تجزیه، $j = 1$ خواهد بود که در این صورت طول پنجره‌ها، به اندازه‌ی دو نمونه از سیگنال می‌باشد. اگر $j = 0$ باشد هر نمونه از سیگنال، یک پنجره خواهد بود. اگر برای تمام پنجره‌هایی که در بزرگترین مقیاس، J ، قرار دارند، این تجزیه را تا کوچکترین مقیاس ادامه دهیم، درخت تجزیه‌ی کامل را خواهیم داشت. اما با توجه به اینکه اگر طول پنجره‌ها خیلی کوتاه باشد، برای آن پنجره‌ها، کاهش نرخ بیت نخواهیم داشت، و همچنین با توجه به این که تقریب به وسیله‌ی توابع چندجمله‌ای، قابلیت تقریب اغلب شکل موج‌ها را دارد، بنابراین به منظور کاهش حجم محاسبات و افزایش سرعت، یک محدوده هم برای حد پایین مقیاس‌ها در نظر می‌گیریم و آن را با J_{min} نشان می‌دهیم. حد بالای مقیاس‌ها را نیز همان طور که در بالا گفته شد با J نشان می‌دهیم. مقادیر J_{min} و J ، در بردار داده‌ی خروجی نوشته می‌شوند تا هنگام بازسازی سیگنال از آن‌ها استفاده شود. شکل ۵-۱ یک نمونه از بخشی از یک درخت تجزیه‌ی کامل را نشان می‌دهد.



شکل ۵-۱. بخشی از یک درخت تجزیه‌ی کامل. در این شکل، هر کدام از گره‌های سطر بالایی به طول 2^J و هر کدام از گره‌های سطر پایینی به طول 2^{Jmin} می‌باشند

۵-۲-۳ - نحوه‌ی تقریب در داخل پنجره‌ها

با توجه به اینکه شکل موج‌های سیگنال‌های صوتی، مشابه توابع چندجمله‌ای است، بنابراین استفاده از توابع چندجمله‌ای برای تقریب شکل موج‌ها می‌تواند گزینه‌ی خوبی باشد. دو مزیت اصلی برای استفاده از توابع چندجمله‌ای برای تقریب به منظور فشرده‌سازی وجود دارد: اول اینکه آن‌ها می‌توانند شکل موج‌های مختلفی را با دقت دلخواه، بسته به مرتبه‌ی چندجمله‌ای تقریب بزنند، و دوم این که آن‌ها با تعداد پارامترهای کمتری بیان می‌شوند که این، برای کاربردهای فشرده‌سازی ضروری می‌باشد. بعضی توابع (مانند توابع منطقی) تعداد پارامترهای بسیار کمی دارند و بنابراین باعث نرخ فشرده‌سازی بیشتر می‌شوند اما فقط توابع خاصی را می‌توانند به خوبی تقریب بزنند. توابع دیگر می‌توانند شکل موج‌های مختلفی را نمایش دهند اما نیاز به پارامترهای بیشتری دارند و بنابراین فشرده‌سازی کمتری دارند [۶۰]. تقریب فریم‌های صوتی با استفاده از توابع چندجمله‌ای با روش مینیمم مربعات خطا، با حل یک دستگاه معادلات نرمال انجام می‌شود [۶۳]. این روش، ضرایب چندجمله‌ای را به ما می‌دهد. اگر فرض می‌کنیم که طول پنجره‌ی مورد نظر N ، و مرتبه‌ی چندجمله‌ای، $order$ ، باشد در این صورت $order + 1$ تا ضریب چندجمله‌ای به دست خواهد آمد. اگر $order < N - 1$ باشد در این صورت تابع چندجمله‌ای، از بعضی نقاط در نزدیکی مقادیر سیگنال عبور می‌کند به طوری که مجموع مربعات اختلافات تمام مقادیر سیگنال و تابع چندجمله‌ای در داخل پنجره‌ی مورد نظر، در مقایسه با تمام توابع چندجمله‌ای ممکن با مرتبه‌ی $order$ ، مینیمم

می‌شود. اثبات این مورد در بخش مربوط به تئوری تقریب چندجمله‌ای آمده است. هر چقدر $order$ کمتر باشد خطای تقریب بیشتر می‌شود [۶۰].

با توجه به این که ما به دنبال موازنه‌ی بین نرخ بیت و اتلاف هستیم پس باید در داخل یک پنجره به طول N ، آن $order$ را که باعث مینیمم شدن تابع هزینه $cost$ می‌شود، به عنوان مرتبه‌ی چندجمله‌ای بهینه در نظر بگیریم و هزینه‌ی مربوط به این مرتبه را، برای پنجره‌ی مذکور در نظر بگیریم.

اما برای رسیدن به هدف فشرده‌سازی و همچنین در کاربردهای عملی نیازی به بررسی تمام $order$ های ممکن، از صفر تا $N - 1$ نیستیم. بنابراین به منظور کاهش حجم محاسبات، می‌توان یک محدودیت برای مرتبه‌های چندجمله‌ای‌ها در نظر گرفت. اما بهتر است این محدودیت را برای حد بالای مرتبه‌ها در نظر بگیریم و حد پایین را که صفر است تغییر ندهیم. زیرا از طرفی محاسبات مربوط به مراتب پایین، کمتر است و از طرف دیگر ممکن است بخش‌هایی از سیگنال (مثل قسمت‌های سکوت) باشند که بتوان آن‌ها را با یک مقدار ثابت (چند جمله‌ای مرتبه‌ی صفر) نمایش داد. بنابراین فریم‌های با طول‌های خیلی بزرگ هم می‌توانند با چندجمله‌ای از مرتبه‌ی صفر نمایش داده شوند.

۵-۲-۴ - موازنه - تابع هزینه

روش پیشنهادی یک روش با اتلاف است. در بحث فشرده‌سازی با اتلاف، دو پارامتر نقش بسیار مهمتری ایفا می‌کنند. نرخ بیت و میزان اتلاف (معیار کیفیت سیگنال بازسازی شده). هدف فشرده‌سازی، رسیدن به نرخ بیت پایین به ازای اتلاف کم است. اما این دو پارامتر همیشه در رقابت هستند. با بهبود یکی، دیگری بدتر می‌شود. ما به دنبال مینیمم کردن هر دو پارامتر هستیم. راه حل برای یافتن جواب بهینه، این است که برای یکی از پارامترها، یک محدودیت در نظر گرفته و به دنبال مینیمم کردن پارامتر بعدی باشیم. یعنی درصد مینیمم کردن D ، باشیم، طوری که نرخ بیت R ، از یک حد مشخص شده R^* ، بیشتر نشود و یا نرخ بیت کل سیگنال را در شرایطی به مینیمم مقدار

خود برسانیم که اتلاف سیگنال بازسازی شده، از یک حد از قبل مشخص شده، D^* ، بیشتر نشود. معمولاً در این گونه موارد از تابع هزینهی لاگرانژ $cost = D + \lambda R$ [۶۴] برای حل مسأله استفاده می‌شود. در صورت استفاده از این رابطه دیگر نیازی به تعریف محدودیت‌های مذکور نیست. در این جا ثابت λ ، بین نرخ بیت، R ، و اتلاف، D ، موازنه ایجاد می‌کند. با افزایش λ ، نرخ بیت کاهش یافته و اتلاف زیادتر می‌شود (اهمیت نرخ بیت بیشتر می‌شود). پس نرخ بیت و اتلاف، وابسته به λ می‌شوند.

واضح است که اگر ما مینیمم تابع هزینهی $cost$ را به ازای $\lambda = \lambda_0$ پیدا کنیم، در واقع می‌نیمم اتلاف را برای حداکثر نرخ بیت مجاز $R(\lambda) = R^*$ ، و یا مینیمم نرخ بیت را برای حداکثر اتلاف مجاز $D(\lambda) = D^*$ پیدا کرده‌ایم. در ادامه، الگوریتم پیدا کردن درخت تجزیه‌ی بهینه براساس معیار نرخ - اتلاف را توضیح می‌دهیم و بعد از آن به نحوه‌ی تعیین معیار اتلاف و نرخ بیت و نحوه‌ی تقریب خواهیم پرداخت.

در ادامه، الگوریتم پیدا کردن درخت تجزیه‌ی بهینه بر اساس معیار نرخ-اتلاف را توضیح می‌دهیم و بعد از آن به نحوه‌ی تعیین معیار اتلاف و نرخ بیت و .. خواهیم پرداخت.

۵-۲-۵- الگوریتم پیدا کردن درخت تجزیه‌ی بهینه

در زیر الگوریتم بهینه‌سازی درخت تجزیه را به صورت پایین به بالا توضیح می‌دهیم. در این جا علامت φ نماینده‌ی درخت تجزیه است. الگوریتم به صورت زیر است.

- درخت تجزیه‌ی کامل را تشکیل بده (φ_0) ، یعنی سیگنال را در سطوح تجزیه‌ی $J = J_{min}, \dots, J$ ، به پنجره‌های غیر هم پوشانی با طول‌های 2^j افراز کن.

- برای هر فریم f (هرگره از درخت تجزیه) مقادیر زیر را حساب کن:

$$c_f = \min(cost) \quad \circ$$

- $order_f$ و $coefs_f$ ، به ترتیب، مرتبه و ضرایب چندجمله‌ای که به ازای آن، c_f حاصل شد.

- $\varphi = \varphi_0$. به این معنی که فعلا، خروجی، برابر با سیگنال تقریب زده شده در سطح J_{min} می باشد. یعنی تمامی پنجره‌های بهینه، فعلا، به طول $2^{J_{min}}$ می باشند.

$$j = j_{min} + 1 \quad \bullet$$

- حلقه: برای هر فریم f در سطح j :

$$s_f = \sum_i m_{f_i} \quad \circ \quad (\text{مجموع هزینه‌ی تمام زیر فریم‌های فریم جاری})$$

$$m_f = \min(s_f, c_f) \quad \circ$$

- اگر $m_f = c_f$ ، فریم f را به همراه مرتبه و ضرایب چندجمله‌ای مربوط نگه

داشته و تمام زیر فریم‌های آن را از درخت تجزیه حذف کن.

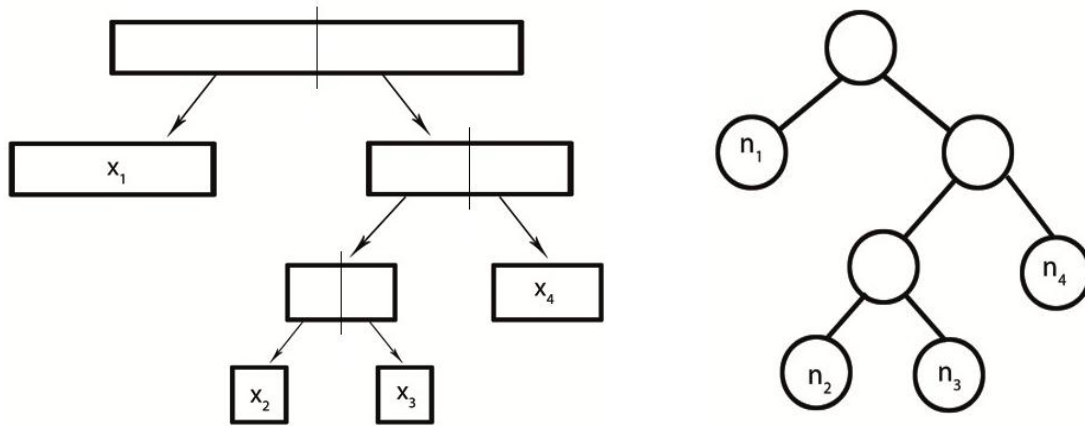
- اگر $m_f = s_f$ ، همان زیر فریم‌های قبلی را به عنوان گره‌های بهینه‌ی فعلی

در نظر بگیر و فریم f را از بررسی‌های بعدی خارج کن.

$$j = j + 1 \quad \bullet$$

- اگر $J \leq j$ به ابتدای حلقه برو.

در پایان الگوریتم، درخت تجزیه‌ی بهینه را خواهیم داشت. شکل ۲-۵ یک مثال از یک بخشی از افراز فریم‌ها و درخت تجزیه‌ی مربوطه را نشان می‌دهد.



شکل ۲-۵. یک مثال از درخت تجزیه‌ی بهینه را نشان می‌دهد. در این جا، بعد از اتمام الگوریتم، گره‌های n_1 ، n_2 ، n_3 و n_4 ، که معادل قسمت‌های x_1 ، x_2 ، x_3 و x_4 از سیگنال هستند، به عنوان گره‌های بهینه انتخاب شده‌اند.

۵-۲-۶- تعیین معیار اتلاف (D)

تعیین معیار اتلاف برای هر کدام از پنجره‌ها، به منظور انتخاب مرتبه‌ی چندجمله‌ای مناسب برای تقریب، بستگی به معیار کیفیت سیگنال بازسازی شده دارد. یکی از مهمترین و معمول‌ترین معیارهای ارزیابی عملکرد الگوریتم‌های فشرده‌سازی، سیگنال به نویز است [۴۷،۴۱].

فرض می‌کنیم، S ، سیگنال اصلی به طول N بوده و s_1, s_2, \dots, s_N ، نمونه‌های آن باشند. سیگنال بازسازی شده را با \hat{S} و نمونه‌های آن را با $\hat{s}_1, \hat{s}_2, \dots, \hat{s}_N$ نشان می‌دهیم. همچنین فرض می‌کنیم پس از اتمام الگوریتم بهینه سازی در روش مبتنی بر وجلت، فریم‌ها یا گره‌های نهایی انتخاب شده برای درخت تجزیه‌ی بهینه، f_1 و f_2 و و f_M باشند. هر کدام از این فریم‌ها می‌توانند متعلق به سطوح تجزیه‌ی مختلف از J_{min} تا J باشند. طول هر فریم f_i ، n_i می‌باشد. در این صورت اگر هدف نهایی ما رسیدن به بیشترین سیگنال به نویز کلی باشد داریم:

$$\begin{aligned}
\arg \max_{\varphi}(SNR) &= \arg \max_{\varphi} \left(10 \log_{10} \frac{\sum_{n=1}^N s_n^2}{\sum_{n=1}^N (s_n - \hat{s}_n)^2} \right) \\
&= \arg \max_{\varphi} \left(\frac{\sum_{n=1}^N s_n^2}{(s_1 - \hat{s}_1)^2 + (s_2 - \hat{s}_2)^2 + \dots + (s_n - \hat{s}_n)^2} \right) \\
&= \arg \max_{\varphi} \left(\frac{\sum_{n=1}^N s_n^2}{\sum_{i=1}^M \sum_{j=1}^{n_i} (f_i(j) - \hat{f}_i(j))^2} \right) \\
&= \arg \min_{\varphi} \left(\sum_{i=1}^M E(f_i - \hat{f}_i) \right) \tag{۷-۵}
\end{aligned}$$

که در آن، $E(\cdot)$ ، به معنی انرژی است. بنابراین نتیجه می‌گیریم که برای رسیدن به SNR بالا در کل سیگنال بازسازی شده‌ی کامل، باید در تک تک فریم‌های کوچک در اندازه‌ها و مقیاس‌های مختلف، درصد مینیمم کردن مجموع مربعات خطای بردار f_i و \hat{f}_i باشیم. پس بنابراین در تابع هزینه‌ی $cost = D + \lambda R$ ، مقدار D را برابر انرژی اختلاف سیگنال داخل فریم و نمایش تقریب چندجمله‌ای آن، قرار می‌دهیم یعنی:

$$D = E(f - \hat{f}_i) = \sum_{j=1}^{n_i} (f_i(j) - \hat{f}_i(j))^2 \tag{۸-۵}$$

بنابراین حالا در می‌یابیم که استفاده از روش مینیمم مربع خطا جهت پیدا کردن ضرایب چندجمله‌ای (که در بخش تئوری چندجمله‌ای‌ها اثبات شد)، انتخاب مناسب و بهینه می‌باشد.

۵-۲-۷- تعیین معیار نرخ بیت (R)

برای هر گره از درخت تجزیه، چند پارامتر ارسال می‌شوند. یکی از این پارامترها طول پنجره است که به اندازه‌ی 2^j می‌باشد و $j_{min} \leq j \leq J$. به جای طول پنجره، مقدار j ، را ارسال می‌کنیم. حال اگر $J - j_{min} < 8$ باشد، می‌توان طول پنجره را با سه بیت نشان داد. لازم به ذکر است که در ابتدای بردار داده‌ی فشرده شده‌ی خروجی، مقادیر j_{min} و J ذخیره شده‌اند. نتایج تجربی نشان

می‌دهد که برای $j_{\min} = 3$ و $J = 10$ ، نتایج بهینه حاصل می‌شود و برای J های بزرگتر از ۱۰، بهبود زیادی حاصل نمی‌شود.

پارامتر بعدی، مرتبه چندجمله‌ای است. این پارامتر در اکثر مواقع کمتر از ۲۰ است. همان طور که در قسمت محدود کردن مراتب چندجمله‌ای توضیح داده خواهد شد، ما خودمان مراتب چندجمله‌ای‌ها را به مقداری در حدود ۲۰ و یا کمتر محدود می‌کنیم. از آن جا که اکثر این مراتب، زیر ۱۶ هستند، بنابراین با ۴ بیت قابل نمایش است. در صورتی که مرتبه، بیش از ۱۴ باشد، همانطور که در توضیح مرجع [۲] گفته شد از کلمات ۴ بیتی بیشتری برای نمایش آن استفاده می‌شود. یعنی تعداد کلمات تعیین کننده مرتبه به صورت $\left\lfloor \frac{\text{order}+1}{15} \right\rfloor$ است که علامت $\lfloor x \rfloor$ ، کوچکترین عدد صحیح بزرگتر یا مساوی x می‌باشد. پارامتر بعدی که باید ارسال شود ضرایب چندجمله‌ای است. حال اگر فرض کنیم که برای کوانتیزاسیون این مقادیر مربوط به تابع چندجمله‌ای، B بیت استفاده شده است در این صورت می‌توان مقدار R را در تابع هزینه $\text{cost} = D + \lambda R$ ، به صورت (۹-۵) تعریف کرد:

$$R = 3 + \left\lfloor \frac{\text{order} + 1}{15} \right\rfloor \times 4 + (\text{order} + 1) \times B \quad (9-5)$$

که در آن، order مرتبه‌ی چندجمله‌ای است.

۵-۲-۸- کدگذاری ضرایب چندجمله‌ای

یکی از قسمت‌های اصلی الگوریتم‌های فشرده‌سازی، کوانتیزاسیون پارامترها، به منظور ذخیره سازی، کاهش تعداد بیت‌ها، کاهش آنتروپی و... و رسیدن به هدف فشرده‌سازی می‌باشد. اما مشکل این است که ضرایب چندجمله‌ای، متأسفانه دارای محدوده‌ی دینامیکی بسیار بالایی می‌باشند. بنابراین کوانتیزه کردن آن‌ها مشکل ساز است. علاوه از آن، ضرایب مختلف محدوده‌ی دینامیکی مختلفی دارند و این مستلزم آن است که برای کدگذاری هر نوع ضریب تعداد صحیح و مناسب بیت‌ها پیدا شود [۶۵]. برای اجتناب از این مشکلات، اگر بتوانیم یک تبدیل کاملاً برگشت پذیر بین

نمونه‌های سیگنال و تابع چندجمله‌ای پیدا کنیم در این صورت این مشکل حل خواهد شد [۶۰]. پیدا کردن این تبدیل به صورت زیر است. با فرض این که چندجمله‌ای از مرتبه‌ی P باشد، با انتخاب $P + 1$ نقطه از تابع چندجمله‌ای می‌توان کل تابع چندجمله‌ای را به صورت کامل و بدون خطا از روی ضرایب تابع، بازسازی نمود که منجر به درون یابی می‌شود. پس به جای کدگذاری $P + 1$ ضریب چندجمله‌ای که از فریم به طول N حاصل شده‌اند، می‌توانیم دقیقاً $P + 1$ نقطه‌ی دلخواه از تابع چندجمله‌ای را انتخاب کرده و آنها را کدگذاری کنیم. با استفاده از $P + 1$ نقطه‌ی مربوط به تابع چندجمله‌ای می‌توان ضرایب چندجمله‌ای را بازسازی نمود. زیرا یک چندجمله‌ای درجه‌ی P که از $P + 1$ نقطه‌ی مشخص عبور کند، منحصر به فرد است. این مورد، منجر به حل یک دستگاه معادلات $P + 1$ معادله $P + 1$ مجهولی می‌شود که به راحتی قابل حل است. در این صورت علاوه از اینکه از مشکلات مربوط به کوانتیزاسیون ضرایب چندجمله‌ای آسوده شده‌ایم، بلکه می‌توانیم از مزایای وجود همبستگی بین نقاط در حوزه‌ی زمان نیز، برای فشرده‌سازی بیشتر بهره ببریم. با انتخاب محل مناسب این $P + 1$ نقطه، در حوزه‌ی زمان، می‌توانیم بیشتر به فشرده‌سازی کمک کنیم.

قبل از ادامه‌ی بحث لازم است یادآوری کنیم که ما توانستیم از روی $P + 1$ نقطه‌ی دلخواه روی تابع چندجمله‌ای، $P + 1$ ضریب مربوط به آن چندجمله‌ای را بازسازی کنیم و با استفاده از ضرایب چندجمله‌ای می‌توان تابع چندجمله‌ای، یا به عبارت بهتر، سیگنال بازسازی شده را با درون یابی، بدون خطا بدست آورد. اما باید به این نکته توجه شود که هر کدام از $P + 1$ نقطه‌ی دلخواه انتخاب شده، دارای طول و عرض (محل و مقدار) هستند. یعنی به ازای هر نقطه، دو عدد داریم. حال برای اینکه این مشکل نیز حل شود، به این شکل عمل می‌کنیم که با هر الگوریتمی که در زمان کدگذاری، محل این $P + 1$ نقطه را انتخاب کرده‌ایم، موقع کد گشایی نیز با همان الگوریتم محل نقاط را بدست می‌آوریم تا هنگام ذخیره سازی، فقط مقادیر نقاط را ذخیره کنیم. ما در این جا، نقاط ابتدایی و انتهایی فریم را انتخاب کرده و برای باقی نقاط، $P - 1$ نقطه با فواصل مساوی در داخل فریم انتخاب می‌کنیم.

قابل توجه است که با تقریب چندجمله‌ای‌ها توانستیم، سیگنال را فشرده کرده و با تعداد ضرایب کمتری نمایندگی کنیم. اما نکته‌ی بسیار مهم این است که با اعمال ترفند مذکور، اعداد به دست آمده، باز هم در حوزه‌ی زمانی سیگنال هستند. بنابراین می‌توانیم با انتخاب محل مناسب برای این نقاط، از مزایای وجود همبستگی و تزیاید در حوزه‌ی زمان، در جهت کاهش نرخ بیت، بیشتر بهره مند شویم. یکی از این موارد، هنگام کوانتیزاسیون است که باید این نکته را در نظر داشته باشیم.

۵-۲-۹- نحوه‌ی انتخاب نقاط تابع

نحوه‌ی انتخاب نقاط مربوط به تابع، مهم می‌باشد. اگر در انتخاب محل این نقاط دقت شود، می‌توان از مزایایی مثل همبستگی بین مقادیر بردار، استفاده کرد. ما این نقاط را به این شکل انتخاب کرده‌ایم. مقدار نمونه‌های اول و آخر فریم، جزء نقاط انتخابی هستند. حال برای انتخاب محل بقیه‌ی نقاط، آن‌ها را با فواصل مساوی از هم درون فریم انتخاب می‌کنیم، به طوری که در نهایت، فواصل تمام نقاط مربوط به یک فریم (تا حد امکان) با هم برابر هستند. البته به دلیل گسسته بودن محل نقاط، این فواصل نمی‌توانند همیشه کاملاً مساوی باشند.

۵-۲-۱۰- محدود کردن مراتب چندجمله‌ای‌ها

آزمایشات نشان می‌دهد که وقتی که مراتب چندجمله‌ای‌ها از یک حدی بیشتر می‌شود (معمولاً در مقادیر بزرگتر از ۲۲)، کوانتیزه کردن نقاط انتخاب شده از تابع (با تعداد بیت‌های کم)، تأثیر نامطلوبی در بازسازی ضرایب چندجمله‌ای دارد. به عبارت دیگر، چندجمله‌ای بازسازی شده، تخریب می‌شود، و این هم شاید به این دلیل است که وقتی مرتبه‌ی چندجمله‌ای بالاست، انعطاف زیادی داشته و حالت‌های زیادی می‌تواند داشته باشد و شکل موج‌های مختلفی را می‌تواند تقریب بزند. بنابراین با اعمال کوانتیزاسیون روی نقاط تابع، تابع بازسازی شده از روی این نقاط، ممکن است در بعضی نقاط دیگر (غیر از نقاط انتخاب شده) دچار اعوجاج شدید بشود. برای حل این مسأله یا باید تعداد بیت‌های کوانتیزاسیون را افزایش دهیم، و یا این که مراتب این چندجمله‌ای‌ها را محدود کنیم.

آزمایش‌های ما نشان می‌دهد که افزایش تعداد بیت‌های کوانتیزاسیون، در ازای بهبود اندک کیفیت، نرخ فشرده‌سازی را تا حد زیادی افزایش می‌دهد. در بعضی مواقع برای اجتناب از تخریب چندجمله‌ای، باید تعداد بیت‌های کوانتیزاسیون را تا حد ۱۴ بیت یا بالاتر برد. در این صورت تمام چندجمله‌ای‌های دیگر (از مراتب پایین‌تر) که نیاز به بیت‌های کوانتیزاسیون زیاد نداشتند هم با بیت‌های زیاد کوانتیزه می‌شوند، و این مسأله باعث افزایش محسوس نرخ بیت می‌شود. بنابراین لازم است تا مراتب چندجمله‌ای‌ها را محدود کنیم تا از یک حد مشخص بالاتر نروند. با محدود کردن مراتب چندجمله‌ای، سرعت سیستم نیز افزایش می‌یابد. زیرا هم تعداد حالت‌های مقایسه برای یافتن مرتبه‌ی بهینه کاهش می‌یابد، و هم اینکه مراتب بالاتر، محاسبات زیادتری هم دارند.

۵-۲-۱۱- کوانتیزاسیون پارامترها

عمده‌ترین پارامترها در روش پیشنهادی، پارامترهای مربوط به طول هر کدام از فریم‌ها، مرتبه‌ی چندجمله‌ای‌های استفاده شده برای تقریب سیگنال در داخل فریم و مهم‌تر از همه ضرایب چندجمله‌ای‌ها هستند. اما همان طور که قبلاً توضیح دادیم به جای ضرایب چندجمله‌ای از مقادیر خود سیگنال استفاده می‌کنیم. اعداد مربوط به طول فریم‌ها و مرتبه‌ی توابع، اعداد صحیح بوده و نیازی به کوانتیزاسیون ندارند. علاوه از آن این اعداد باید کاملاً بدون خطا کد شوند. بنابراین بحث اصلی در کوانتیزاسیون، مربوط به مقادیر انتخاب شده از تابع است. همان طور که قبلاً نیز توضیح دادیم، در استفاده از مقادیر سیگنال به جای ضرایب چندجمله‌ای می‌توانیم از مزایای همبستگی نمونه‌های سیگنال در حوزه‌ی زمان بهره‌مند شویم. بنابراین در کوانتیزه کردن این مقادیر سعی می‌کنیم تا مؤثرترین روش را برای حذف بیشتر تزیاید و رسیدن به فشرده‌سازی بیشتر انتخاب کنیم. در قسمت توضیحات مربوط به کوانتیزاسیون، سه روش مختلف و معمول را شرح دادیم. روش کوانتیزاسیون یکنواخت، روش DPCM و روش کوانتیزاسیون غیر یکنواخت لگاریتمی ($\mu - law$). گفته می‌شود که روش کوانتیزاسیون غیریکنواخت، در نهایت SNR بهتری نسبت به کوانتیزاسیون یکنواخت دارد [۷]. نتایج آزمایشات ما نشان می‌دهد که این افزایش SNR همراه با افزایش آنتروپی

نمادهای خروجی چندی‌ساز نیز هست. بنابراین در صورتی که بعد از کوانتیزاسیون بخواهیم از کدگذاری آنتروپی استفاده کنیم، در این صورت کوانتیزاسیون یکنواخت عملکرد بهتری دارد. روش DPCM هم همان طور که قبلاً گفته شد، از تزیاید و همبستگی بین نمونه‌ها بهره می‌گیرد. در این جا، ما هر سه روش را امتحان کردیم تا نتایج هر سه را ببینیم.

۵-۲-۱۲- کدگذاری آنتروپی

برخلاف روش‌هایی مثل DPCM و ADPCM که تزیاید را با بهره‌گیری از همبستگی موجود در سیگنال حذف می‌کنند، روش‌های کدگذاری آنتروپی، از احتمال وقوع نمادها بهره می‌گیرند [۱۱]. بنابراین اگرچه در استفاده از DPCM، همبستگی سیگنال حذف می‌شود اما آنتروپی نمونه‌ها حتی کمتر هم می‌شود (چون خروجی DPCM، در صورت وجود همبستگی کافی بین نمونه‌ها، محدوده‌ی دینامیکی پایین دارند). بنابراین استفاده از یک کدگذاری آنتروپی، می‌تواند تأثیر بیشتری داشته باشد.

با توجه به این که در این جا مقادیر تکراری متوالی زیادی نداریم بنابراین روش کدگذاری دوره‌ی تداوم نمی‌تواند مفید باشد. در بین روش‌های هافمن و کدگذاری حسابی نیز، چون کدگذاری حسابی، نسبت به هافمن، بهینه‌تر است [۱۰]، بنابراین ما نیز، هم در این جا و هم در روش‌های مبتنی بر تبدیل از آن استفاده کرده‌ایم. نتایج پیاده‌سازی‌های ما هم بهتر بودن کدگذار حسابی را نسبت به هافمن، مخصوصاً وقتی که طول بردار کد شونده بزرگ باشد، تأیید می‌کند. برای سیگنال‌های طولانی با تعداد نمادهای کم و توزیع غیر یکنواخت، کدگذار حسابی بهتر از هافمن فشرده می‌کند [۱۴]. که می‌توان گفت خروجی DPCM، به شرط اینکه سیگنال ورودی آن به قدر کافی بین نمونه‌هایش همبستگی داشته باشد، این شرایط را دارد.

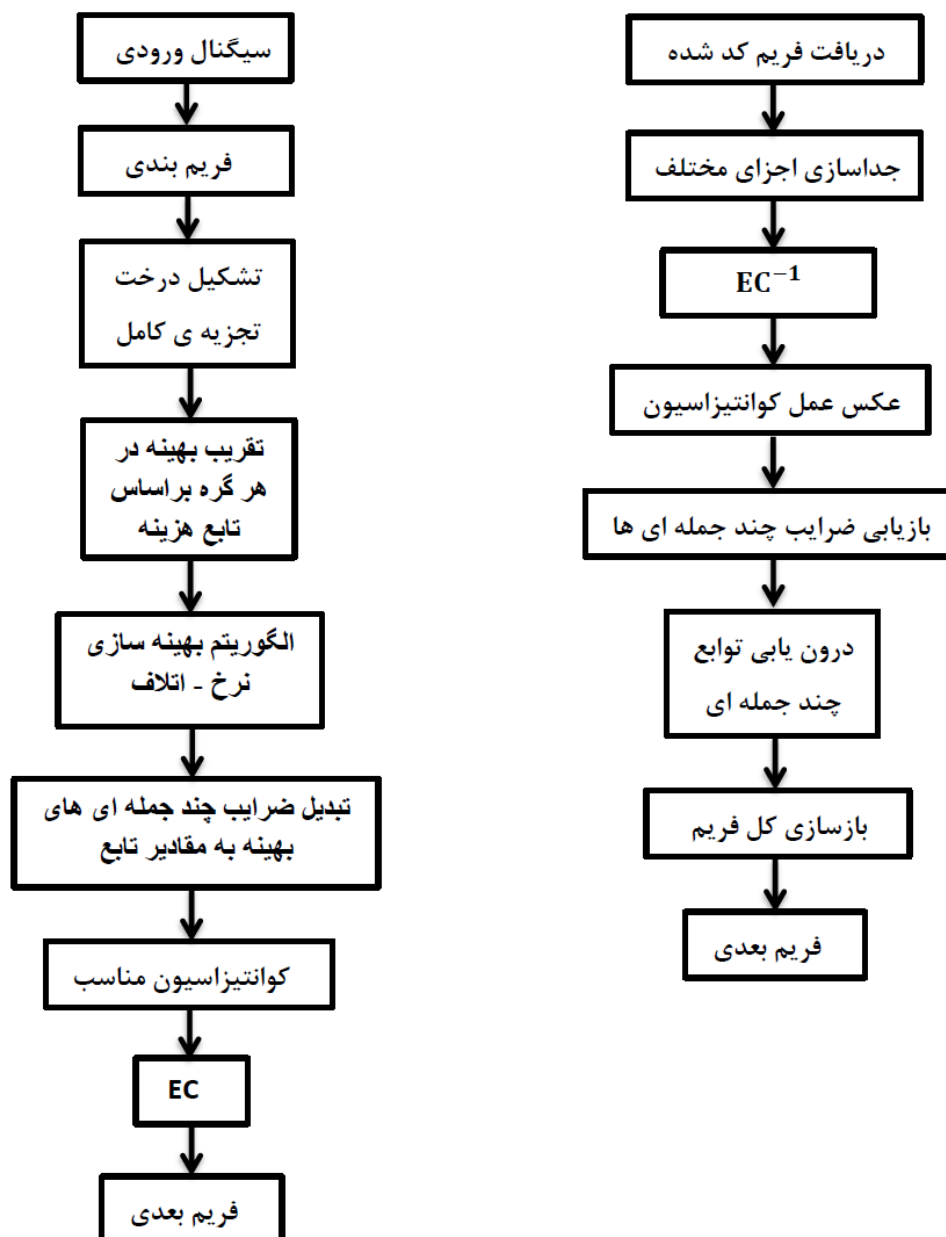
۵-۲-۱۳- روند نما

روندنمای روش ارائه شده به صورت خلاصه در شکل ۵-۳ نشان داده شده است. سیگنال ورودی وارد می‌شود. هر کدام از مقادیر J و B را با چهار بیت وارد بردار خروجی می‌کنیم. B تعداد بیت‌های کوانتیزاسیون است. ابتدا فریم‌بندی انجام می‌شود. سپس در هر فریم درخت تجزیه‌ی کامل را تشکیل داده و بعد از تقریب هر کدام از گره‌های درخت تجزیه با بهینه‌ترین تابع، الگوریتم بهینه سازی نرخ - اتلاف را اجرا می‌کنیم. حال درخت تجزیه‌ی بهینه و مراتب چندجمله‌ای‌های گره‌های بهینه را در بردار خروجی به صورت مجزا وارد می‌کنیم. در هنگام بازسازی سیگنال با داشتن طول فریم، با ورود رشته‌ی کد مربوط به درخت تجزیه (طول فریم یا گره‌های بهینه) و کدگشایی تدریجی رمز مربوط به درخت تجزیه، نقطه پایان رمز مربوط به درخت تجزیه مشخص می‌شود و نیاز به اطلاعات اضافی نیست. این نقطه زمانی مشخص می‌شود که مجموع طول فریم‌های (گره‌های) بهینه‌ی درخت تجزیه، برابر طول فریم اصلی باشد. از این جا تعداد این گره‌ها نیز مشخص می‌شود و بنابراین ادامه‌ی رشته‌ی باینری کد، مربوط به مراتب چندجمله‌ای‌ها خواهد بود و از روی تعداد گره‌های بهینه نیز نقطه‌ی پایان کد مربوط به مراتب هم مشخص می‌شود. در ادامه، باید کد مربوط به ضرایب چندجمله‌ای‌ها را بفرستیم. همان طور که قبلاً نیز توضیح دادیم، به جای این ضرایب، به همان تعداد از مقادیر تابع را در مکان‌هایی مشخص انتخاب کرده و آن‌ها را ارسال می‌کنیم. این مقادیر باید کوانتیزه شوند. با توجه به همبستگی بین مقادیر، نتایج روش DPCM امید بخش می‌باشد.

کوانتیزاسیون با B بیت انجام می‌شود. مقادیر M_{min} و M_{max} مربوط به کوانتیزاسیون را بعد از کوانتیزه کردن عددی، در رشته‌ی خروجی وارد می‌کنیم تا هنگام انجام عکس کوانتیزاسیون مورد استفاده قرار گیرند.

نمادهای خروجی کوانتیزاسیون را بعد از اصلاحات لازم به ورودی کدگذاری حسابی می‌دهیم. برای رمز گشایی رمز خروجی از کدگذاری حسابی، باید علاوه از این رشته‌ی رمز، مقادیر احتمال

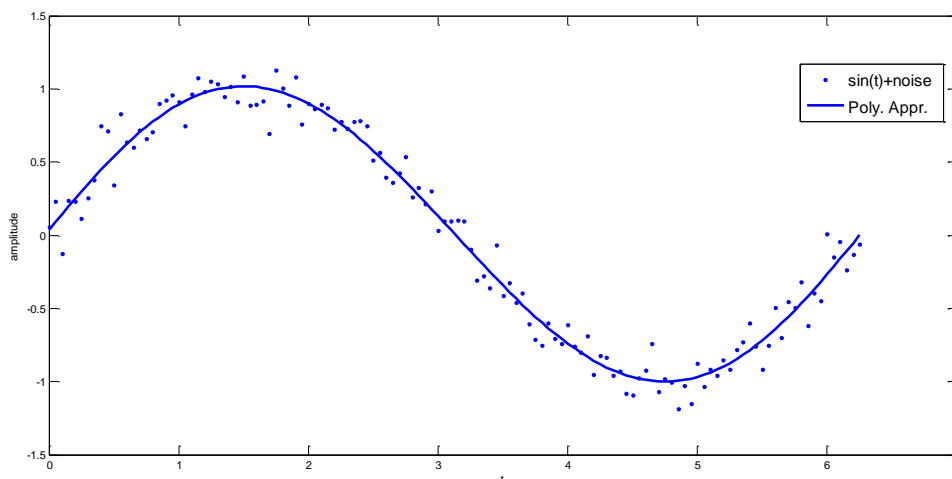
نمادها (تعداد هر نماد در بردار ورودی) و همچنین طول رشته‌ی نمادهای ورودی کدگذاری حسابی را نیز در دست داشته باشیم. طول رشته‌ی نمادهای ورودی کدگذاری حسابی (رشته نمادهای خروجی کوانتیزاسیون) از روی تعداد گره‌ها و همچنین مراتب چندجمله‌ای‌های متناظر هر گره بدست می‌آید و نیازی به وارد کردن آن در رشته‌ی خروجی نیست. اما تعداد هر نماد را باید در رشته‌ی خروجی وارد کنیم.



شکل ۵-۳. روند نمای کلی روش پیشنهادی

۵-۲-۱۴- قابلیت حذف نویز

یکی از مزایای تقریب چندجمله‌ای به روش مینیمم مربع خطا، این است که وقتی که سیگنال نویزی می‌باشد، سیگنال تقریب، سیگنال اصلی نویزی را به صورتی نرم‌تر^۱ تقریب می‌زند. یعنی چندجمله‌ای تلاش می‌کند تا بهترین چندجمله‌ای از درجه‌ی P را طوری پیدا کند که مربع خطا را در طول کل فریم می‌نیمم کند. به عنوان مثال، شکل ۴-۵ یک سینوسی را که نویز گوسی با انحراف معیار ۰/۱ به آن اضافه شده است، نشان می‌دهد. منحنی ممتد رسم شده، تقریب این سینوسی نویزی با استفاده از یک تابع چندجمله‌ای با مرتبه‌ی ۵ است. مشاهده می‌شود که سیگنال تقریب زده شده بسیار نزدیک به سینوسی بدون نویز است.



شکل ۴-۵. سیگنال سینوسی نویزی و تقریب آن به وسیله‌ی تابع چندجمله‌ای

بنابراین به نظر می‌رسد که این روش در حضور سیگنال‌های نویزی، نتایج خوبی از خود نشان دهد. هم‌چنین به نظر می‌رسد این حذف نویز در استفاده از چندجمله‌ای‌های با مرتبه‌ی پایین مؤثرتر باشد. زیرا با زیاد شدن مرتبه‌ی آن، تابع سعی خواهد کرد تا قسمت‌های نویزی سیگنال را دقیق‌تر نمایندگی کند که در این صورت، نویز حذف نشده، بلکه باقی می‌ماند.

¹ Smooth

۵-۲-۱۵- فشرده‌سازی مبتنی بر نواحی مطلوب

مهمترین هدف فشرده‌سازی، کاهش حجم داده است به طوری که کمترین اتلاف اطلاعات از سیگنال داشته باشیم. اما ارزش و اهمیت همه‌ی اطلاعات سیگنال، به یک اندازه نیست. بعضی از قسمت‌های سیگنال نسبت به قسمت‌های دیگر اهمیت بیشتری دارند. در فشرده‌سازی مبتنی بر نواحی مطلوب، سعی بر این است تا نواحی کم اهمیت و با اهمیت (مطلوب) سیگنال شناسایی شده و اتلاف بیشتر در قسمت‌های کم اهمیت صورت بگیرد. در سیگنال‌های صوت و صحبت قسمت‌های سکوت را می‌توان جزء قسمت‌های کم اهمیت در نظر گرفت و بنابراین با اتلاف بیشتر و فشرده‌سازی بیشتر در این مناطق، به بهبود نرخ فشرده‌سازی نهایی کمک کرد. این مسأله مخصوصاً در مورد سیگنال‌های صحبت بیشتر به چشم می‌خورد. زیرا در سیگنال‌های صحبت، قسمت‌های سکوت زیادی داریم.

برای تشخیص و جدا کردن قسمت صحبت و غیر صحبت و سکوت یک بحث گسترده‌ای به نام VAD وجود دارد که تحقیقات زیادی هم در این زمینه انجام شده است. هدف از آشکارسازی صحبت، طبقه بندی فریم ورودی با عنوان فریم صحبت و یا غیر صحبت است. بعضی از الگوریتم‌های فشرده‌سازی گفتار نیز از این روشها استفاده می‌کنند [۴۷، ۶۶]. در این روشها ابتدا سیگنال ورودی به فریم‌هایی تجزیه می‌شود و سپس هر کدام از این فریم‌ها، به یک طبقه بند ارسال می‌شوند تا کلاسه بندی شوند. بعد از دسته بندی، برای هر کدام، روش متفاوتی برای فشرده‌سازی اتخاذ می‌شود. پارامترهایی که معمولاً برای تصمیم گیری استفاده می‌شوند، انرژی، نرخ گذر از صفر و دوره تناوب *Pitch* هستند. یک آشکارساز ساده می‌تواند فقط از یکی از این پارامترها، برای جداسازی استفاده کند [۴۷]. انرژی معمول‌ترین ویژگی در کاربردی‌های VAD است [۶۷]. به عنوان مثال انرژی فریم می‌تواند به صورت زیر برای آشکارسازی استفاده شوند: اگر انرژی از یک حد آستانه، بیشتر است، صحبت و در غیر این صورت غیر صحبت است. بنابراین چالش طراحی، انتخاب یک حد آستانه مناسب

به وسیله آموزش تعداد زیادی از پایگاه داده است. اما استفاده از فقط یک پارامتر، عملکرد سیستم را محدود می‌کند. بنابراین بهتر است برای بهبود عملکرد آشکارسازی، از بیش از یک پارامتر استفاده کرد. در این صورت محاسبات افزایش می‌یابد. این روشها بعضی معایب دارند. در خیلی موارد یک فریم نمی‌تواند به طور قطعی به عنوان گفتار یا غیر گفتار کلاسه بندی شود. همچنین بعضی فریم‌های گذار وجود دارند (گذر از صحبت به غیر صحبت یا برعکس) که این روشها به خوبی قادر به کلاسه بندی نیستند. اگر یک فریم صحبت به اشتباه غیر صحبت تشخیص داده شود (و یا برعکس) کیفیت سیگنال خروجی مطلوب نخواهد بود و این مسأله باعث بروز بعضی صداها ناهنجار و آزار دهنده می‌شود.

ما در این جا یک روش ساده و مؤثر به منظور فشرده‌سازی مبتنی بر نواحی مطلوب ارائه داده‌ایم. روش به این صورت است که یکی از ویژگی‌های مجزاساز بین نواحی صحبت و سکوت را در تابع $cost$ و در پارامتر λ تأثیر می‌دهیم. اما با توجه به این که در الگوریتم بهینه سازی نرخ - اتلاف، مقیاس‌های مختلف با هم مقایسه می‌شوند، بنابراین ویژگی استخراج شده باید مستقل از طول فریم باشد. به طوری که پارامتر λ برای قسمت‌های صحبت و سکوت مقدار متفاوتی داشته باشد. اگر بتوانیم پارامتر λ را طوری تعیین کنیم که برای قسمت‌های صحبت، مقدار کم داشته و برای قسمت سکوت و یا نویز پس زمینه مقدار زیادی داشته باشد در این صورت برای قسمت‌های صحبت، تأثیر جمله‌ی مربوط به اتلاف (D) در تابع هزینه، بیشتر می‌شود و همچنین برای قسمت‌های سکوت و نویز پس زمینه، با بالا بردن λ در این قسمت‌ها، تأثیر نرخ بیت، زیاد خواهد بود. ویژگی‌های معمول مثل انرژی و یا نرخ گذر از صفر شدیداً وابسته به طول فریم هستند. یک گزینه‌ی مناسب برای این انتخاب می‌تواند، واریانس فریم باشد. در مورد سیگنال‌های صوتی معمولاً با افزایش دامنه (انرژی)، واریانس هم افزایش می‌یابد. یعنی در قسمت‌های صحبت که انرژی فریم‌ها نسبت به فریم‌های سکوت در طول فریم برابر، بیشتر است، واریانس هم نسبت به قسمت‌های سکوت زیاد است. به عبارت دیگر، از آن جا که توزیع نمونه‌ها در سیگنال‌های صوتی، توزیع نرمال است، بنابراین با توجه به ماهیت سیگنال‌های

صوتی، با افزایش دامنه و در واقع پراکندگی بیشتر داده، واریانس نیز افزایش می‌یابد. اما یک مزیت واریانس نسبت به انرژی، این است که واریانس مستقل از طول فریم است. اگر بخواهیم از انرژی برای این کار استفاده کنیم می‌توانیم انرژی را به طول فریم تقسیم کنیم تا نسبت به طول فریم‌ها، حساس نباشد.

حال سؤال این است که نحوه‌ی اعمال این ویژگی استخراج شده به چه شکل باشد؟ با توجه به این که در قسمت‌های مطلوب (صحبت)، مقدار واریانس افزایش می‌یابد پس نحوه‌ی اعمال آن باید به شکل معکوس به پارامتر λ باشد تا با افزایش واریانس مقدار λ ، کاهش یابد. (یعنی $\lambda = \frac{c}{\sigma^2}$) در این صورت تابع هزینه به شکل زیر خواهد بود :

$$\text{cost} = D + \lambda R = D + \frac{c}{\sigma^2} R \quad (10-5)$$

که در آن σ^2 ، واریانس فریم و c ، یک عدد ثابت است. در این صورت، به هر میزان که مقدار σ^2 افزایش یابد (قسمت‌های صحبت)، ضریب R ، کاهش یافته و تأثیر D در بهینه‌سازی بیشتر می‌شود.

یک مزیت دیگر این روش این است که تأثیر واریانس به صورت نسبی است. یعنی در یک سیگنال صحبت، واریانس قسمت‌های گفتار نسبت به قسمت‌های سکوت، بیشتر است. توجه داریم که سیگنال‌های صحبت دامنه‌های متفاوت و واریانس‌های متفاوتی دارند و واریانس، بستگی به خود سیگنال دارد. بنابراین در استفاده از این روش، در مورد تمامی سیگنال‌های صحبت، و در تمامی نرخ بیت‌ها، قسمت‌های صحبت نسبت به قسمت‌های دیگر با کیفیت بهتر و نرخ بیت بیشتری فشرده می‌شوند. بنابراین در این روش مشکلات مربوط به مقدار آستانه‌گذاری برای جداسازی فریم‌های گفتار و سکوت نخواهد بود. در این صورت مخصوصاً در نرخ بیت‌های پایین که اتلاف بیشتری داریم، این روش، بیشتر اتلاف را به قسمت‌های سکوت اختصاص می‌دهد.

با وجود این، بعضی از قسمت‌های سیگنال گفتار هستند که دامنه و در نتیجه واریانس نسبتاً کم داشته ولی مهم هستند. برخی از حروف مانند «س»، اگر آرام‌تر ادا شوند، ممکن است دامنه‌ی کمی داشته باشند. همچنین در مواقعی که از حالت سکوت به صحبت گذر می‌کنیم و یا در گذر از صحبت به سکوت، دامنه‌ی سیگنال کمی داریم. در این گونه مواقع، سیگنال اندکی نویزی شنیده می‌شود اما در عوض قسمت صحبت سیگنال، با کیفیت بهتر شنیده می‌شود.

فصل ششم

نتایج

۶-۱- پایگاه داده

روش پیشنهادی در این پایان نامه، برای اغلب سیگنال‌ها حتی سیگنال‌های غیرصوتی هم می‌تواند مورد استفاده قرار گیرد. پایگاه داده‌ی مورد استفاده در این پایان نامه، هم شامل سیگنال‌های صوتی و هم صحبت است. سیگنال‌های صوتی، نه سیگنال هستند که از پایگاه داده‌ی [۲۷]SQAM گرفته شده‌اند. سیگنال‌های صوتی طوری انتخاب شده‌اند که شامل انواع مختلف صداهای موجود در این پایگاه داده باشند. این پایگاه داده، شامل شش سیگنال صحبت نیز هست که دو مورد از آن‌ها به زبان انگلیسی (یکی صدای زن و یکی مرد)، دو مورد به زبان فرانسوی و دو مورد هم به زبان آلمانی است. اما از آن جا که ما از معیار MOS استفاده کرده‌ایم، بنابراین بهتر است تا شنوندگان، با زبان سیگنال‌های صحبت آشنایی داشته باشند. زیرا به عنوان مثال مواردی پیدا می‌شود که سیگنال بازسازی شده، نویزی نیست اما بعضی از صداها و حروف کلمات حذف شده‌اند. در این گونه مواقع و موارد مشابه، نظرات شنوندگانی که آشنا به زبان سیگنال نیستند، نمی‌توانند زیاد معتبر باشند. بنابراین برای سیگنال صحبت، ما از شش سیگنال به زبان فارسی استفاده کرده‌ایم. این شش سیگنال مربوط به شش شخص مختلف می‌باشند که سه تای آن‌ها صدای زن و سه مورد، صدای مرد هستند. تعداد نمونه‌های تمامی سیگنال‌ها مضربی از 2^{11} است. تمامی سیگنال‌ها با فرکانس 44.1 KHz نمونه‌برداری شده و با دقت ۱۶ بیت به صورت یکنواخت کوانتیزه شده‌اند. پایگاه داده‌ی مورد استفاده در این پایان نامه، همراه با مشخصات سیگنال‌ها در انتهای پایان نامه آورده شده است.

۶-۲- معیارهای ارزیابی

در ارزیابی عملکرد سیستم‌های فشرده‌سازی، دو پارامتر اهمیت خاصی دارند. نرخ بیت و کیفیت سیگنال بازسازی شده.

ارزیابی کیفیت سیگنال‌های بازسازی شده، یک بحث مهم و در عین حال مشکل می‌باشد. ارزیابی کیفیت به دو شکل کمی و کیفی انجام می‌شود. معیارهای کمی شامل مقایسه‌ی شکل موج‌ها،

سیگنال به نویز (SNR)^۱، و مقایسه‌ی بعضی ویژگی‌های حوزه‌ی فرکانس می‌شود. معیارهای کیفی، شامل قابلیت فهم و کیفیت ادراکی گوش انسان می‌شود. نوع سومی از معیارهای ارزیابی وجود دارد که سعی می‌کند با روش‌های کمی و نظری، کیفیت ادراکی سیگنال را اندازه بگیرد. این نوع معیارها بحث جدیدی بوده و هنوز به طور کامل گسترش نیافته‌اند [۴۲]. از انواع این روش‌ها می‌توان به PESQ^۲ برای ارزیابی صحبت، و PEAQ^۳ برای ارزیابی کیفیت سیگنال‌های صوتی اشاره کرد. این روش‌ها طوری طراحی شده‌اند که سعی می‌کنند معیار MOS^۴ را تقریب بزنند. اما هنوز نتایج آن‌ها متفاوت از نتایج MOS است. گفته می‌شود که بهتر آن است که PESQ و PEAQ را همراه با معیارهای دیگری مثل MOS به کار برد [۳۹]. از آن جا که کیفیت سیگنال‌های بازسازی شده، بیشتر بر پایه‌ی ادراک است، لذا روش‌های ارزیابی کیفی، بیشتر قابل اطمینان هستند. بنابراین برای ارزیابی کیفیت‌ها، لازم است که به سیگنال گوش دهیم.

ما در این پایان نامه، از هر دو روش کمی و کیفی برای ارزیابی استفاده کرده‌ایم. معمول‌ترین

معیار کمی، سیگنال به نویز (SNR) می‌باشد [۴۱]، و از رابطه‌ی زیر قابل محاسبه است:

$$SNR = 10 \log_{10} \left\{ \frac{\sum_{n=0}^{M-1} s^2(n)}{\sum_{n=0}^{M-1} (s(n) - \hat{s}(n))^2} \right\} \quad (1-6)$$

که در آن، $s(n)$ سیگنال اصلی و $\hat{s}(n)$ سیگنال بازسازی شده می‌باشد. M طول سیگنال است.

معتبرترین معیارهای ارزیابی معیارهای کیفی هستند که در آن از تعدادی شنونده‌ی انسانی

برای نظردهی استفاده می‌شود. MOS یک معیار کیفی است که به وفور برای تعیین کیفیت

سیگنال‌های فشرده شده، استفاده می‌شود [۴۱]. در این روش یک عدد به کیفیت سیگنال بازسازی

شده تخصیص داده می‌شود. سیگنال اصلی، امتیاز ۵ دارد. از شنوندگان خواسته می‌شود تا پس از

گوش کردن به سیگنال، یک امتیاز از ۱ تا ۵، مطابق با جدول ۶-۱، به سیگنال بدهند. امتیاز MOS

¹ Signal to Noise Ratio

² Perceptual Evaluation of Speech Quality

³ Perceptual Evaluation of Audio Quality

⁴ Mean Opinion Score

مربوط به یک سیگنال خاص، میانگین امتیازهایی است که هر کدام از شنوندگان به سیگنال داده‌اند. شنوندگان باید از روشی که برای فشرده‌سازی سیگنال استفاده شده، بی‌اطلاع باشند تا امتیازها اعتبار بیشتری داشته باشند. تعداد شنوندگان در MOS معمولاً بین ۱۲ تا ۲۴ نفر انتخاب می‌شوند [۴۱].

جدول ۶-۱. امتیازها در MOS

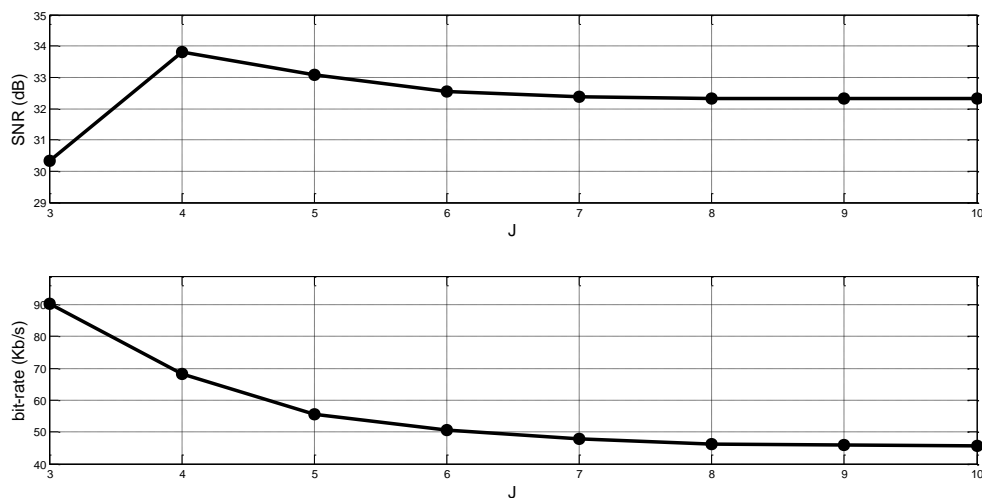
امتیاز	توصیف
۵	عالی (غیر قابل تشخیص)
۴	خوب (قابل تشخیص اما غیر آزار دهنده)
۳	متوسط (کمی آزار دهنده)
۲	ضعیف (آزار دهنده)
۱	بد (خیلی آزار دهنده)

پارامتر مهم دیگر، نرخ بیت است. هر چه میزان فشرده‌سازی بیشتر باشد، نرخ بیت کم خواهد بود. نرخ بیت برابر تعداد بیت‌هایی است که باید در یک ثانیه ارسال شود تا سیگنال بتواند به صورت بلادرنگ ارسال شود. به عبارت دیگر، متوسط تعداد بیت‌های سیگنال فشرده شده برای هر ثانیه از سیگنال است. این تعریف، زمانی است که نرخ بیت با واحد بیت بر ثانیه بیان شود. می‌توان نرخ بیت را با واحدهای دیگر مثل بیت بر نمونه، بیت بر فریم، فریم بر ثانیه و ... بیان کرد.

۶-۳- تعیین مقدار J

هر چقدر که J را بیشتر بگیریم، طول فریم‌ها بیشتر شده، درخت تجزیه بزرگتر و سرعت برنامه کمتر می‌شود. اما از طرفی ممکن است به دلیل الگوریتم بهینه‌سازی، جواب بهتر شود. در این آزمایش سعی داریم تا با آزمایش تعدادی از سیگنال‌های پایگاه داده برای J های مختلف، تأثیر پارامتر J را مشاهده کرده و مقدار مناسبی از لحاظ سرعت و دقت انتخاب کنیم. در این آزمایش، $\lambda = 9.4 \times 10^{-4}$ و $j_{min} = 2$ در نظر گرفته شده و تنها پارامتری که تغییر می‌کند، J است.

همچنین این آزمایش با کوانتیزاسیون DPCM انجام شده است. آزمایش‌ها روی چهار سیگنال `saxophone.wav`، `s6.wav`، `s3.wav` و `piano.wav` انجام شده و میانگین نتایج روی نمودار نشان داده می‌شود. با توجه به این که مقدار ضریب موازنه، تا حدودی پایین است، این جا اهمیت کیفیت سیگنال در تابع هزینه بیشتر مطرح می‌شود. شکل ۶-۱، نتایج این آزمایش را نشان می‌دهد. مقادیر نمودار، میانگین نتایج حاصل از سیگنال‌های تحت آزمایش است.



شکل ۶-۱. بررسی تأثیر مقدار J بر روی سیگنال به نویز و نرخ بیت

نتایج نشان می‌دهد که تغییرات SNR در J های مختلف پایین است. یعنی در تمام طول فریم‌ها، می‌توان به SNR قابل قبول دست یافت. اما باید نرخ بیت را هم در نظر گرفت. مشاهده می‌کنیم که با افزایش J ، نرخ بیت کاهش می‌یابد، و شدت این کاهش در ابتدا زیاد بوده و رفته رفته از شیب منحنی کاسته می‌شود. نتایج این نمودار نشان می‌دهد که برای J های بیشتر از ۹، بهبود عملکرد سیستم ناچیز است. این در حالی است که با افزایش J ، سرعت سیستم کاهش می‌یابد. بنابراین انتخاب $J = 9$ ، یا $J = 10$ ، به نظر مناسب می‌آید. کاهش اندک SNR، برای J های بزرگتر از ۴، می‌تواند به دلیل موازنه‌ی بین نرخ بیت و SNR باشد.

۶-۴- نوع سیگنال

روش‌های مختلف فشرده‌سازی ممکن است برای سیگنال‌های خاصی جواب بهتری داشته و برای بعضی دیگر عملکرد خوبی نداشته باشند. در مقابل، بعضی از روش‌ها مثل کدینگ سینوسی، برای تمام انواع سیگنال‌ها مناسب هستند [۱۷]. پایگاه داده‌ی ما متشکل از انواع مختلف سیگنال‌های صوتی است. در این آزمایش سعی شده تا عملکرد روش مبتنی بر وجلت را بر روی سیگنال‌های مختلف از انواع مختلف آزمایش کنیم و ببینیم که آیا عملکرد روش ارائه شده به نوع سیگنال بستگی دارد و مثلاً برای نوع خاصی از سیگنال‌ها عملکرد بهتری نسبت به بقیه دارد یا این که برای تمام سیگنال‌ها می‌تواند کاربرد داشته باشد. نتایج این آزمایش در جدول ۶-۲ آمده است.

در این آزمایش $\lambda = 9.4 \times 10^{-4}$ ، $J = 10$ و کوانتیزاسیون DPCM، انتخاب شده است. تعداد بیت‌های کوانتیزاسیون، هشت بیت است. نتایج این جدول نشان می‌دهند که روش مبتنی بر وجلت، برای تمامی انواع سیگنال‌های صوتی، جواب مشابهی دارد. به این معنی که به سختی می‌توان یکی از انواع سیگنال‌ها را برای روش پیشنهادی، مناسب‌تر دانست. دلیل این امر را می‌توان در قابلیت چندجمله‌ای‌ها در تقریب انواع مختلف شکل موج‌ها، و همچنین الگوریتم بهینه سازی نرخ-اتلاف دانست. بنابراین از این روش هم در سیگنال‌های صوت و هم صحبت می‌توان استفاده کرد. با این وجود، در مقایسه با دو روش دیگر، نتایج روش پیشنهادی در مورد سیگنال‌های صحبت بهتر از دو روش دیگر است. در مورد سیگنال‌های صوتی نیز بسته به سیگنال مورد نظر، نتایج متفاوت هستند. به نظر می‌رسد فشرده کردن سیگنال‌های موسیقی همراه با آواز، سخت‌تر می‌باشد.

با این توصیف، می‌توان گفت که روش پیشنهادی برای تمام سیگنال‌های صوت و صحبت،

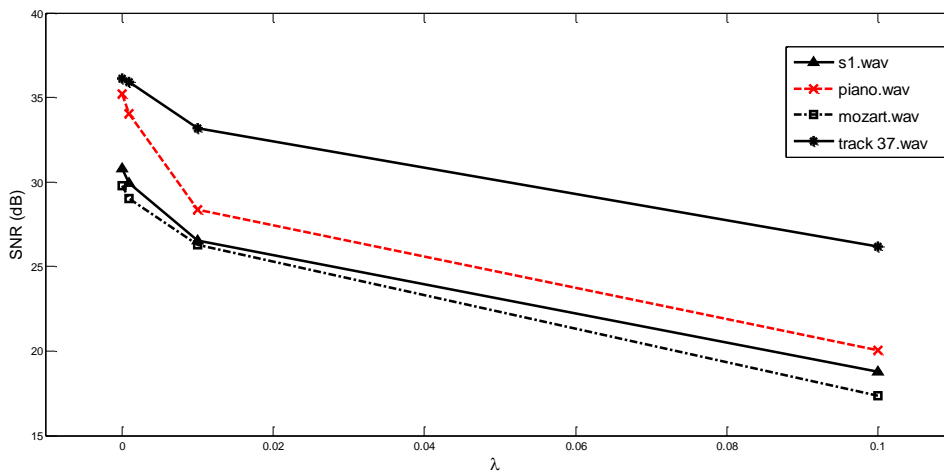
می‌تواند مورد استفاده قرار گیرد.

جدول ۶-۲. نتایج بر روی سیگنال‌های مختلف پایگاه داده

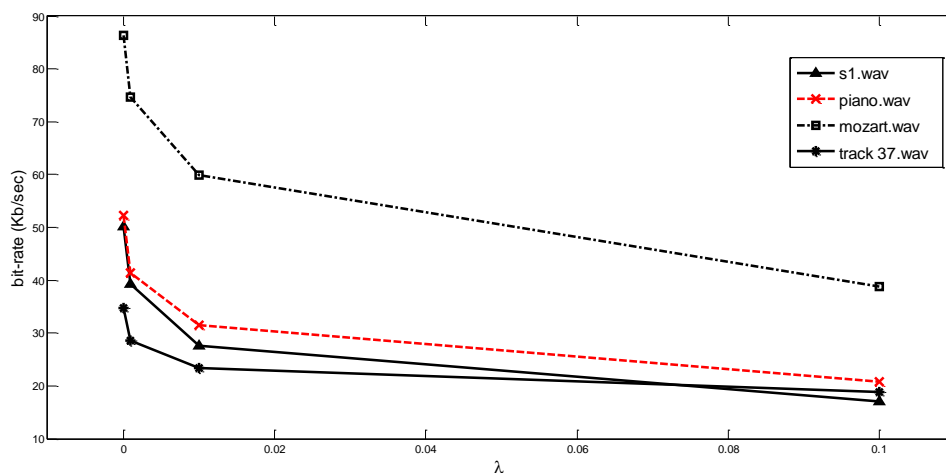
سیگنال	روش مبتنی بر وولت		روش DWT		روش DCT	
	SNR (dB)	نرخ بیت (Kbps)	SNR (dB)	نرخ بیت (Kbps)	SNR (dB)	نرخ بیت (Kbps)
s1.wav	۲۹/۹۳	۳۹/۳۷	۳۱/۵۷	۵۶/۷۷	۳۰/۵۴	۴۸/۹۰
s2.wav	۳۲/۴۷	۴۰/۰۷	۳۲/۶۱	۵۳/۶۰	۳۱/۰۸	۵۱/۴۰
s3.wav	۲۶/۸۱	۵۴/۱۷	۳۰/۲۳	۵۴/۵۲	۲۹/۱۳	۴۶/۱۲
s4.wav	۳۲/۸۶	۴۸/۴۰	۳۲/۶۰	۵۶/۲۳	۳۰/۲۹	۴۶/۴۶
s5.wav	۳۲/۵۰	۵۱/۹۰	۳۲/۰۳	۵۵/۸۳	۳۰/۱۴	۴۶/۸۵
s6.wav	۳۲/۲۸	۴۲/۷۱	۳۲/۴۲	۵۱/۰۳	۳۱/۹۷	۴۸/۱۱
Violon	۳۶/۴۱	۷۲/۱۵	۲۸/۷۰	۵۶/۸۲	۲۸/۵۳	۲۳/۰۵
Piano	۳۴/۰۳	۴۱/۴۲	۳۱/۰۹	۴۷/۰۹	۳۰/۱۳	۲۰/۹۲
Castanet	۲۰/۵۶	۶۰/۰۵	۲۳/۰۴	۴۷/۶۱	۲۳/۲۷	۴۷/۲۵
Saxophone	۳۶/۱۴	۴۴/۳۶	۲۶/۲۳	۵۲/۹۹	۳۰/۴۵	۳۶/۰۹
Track 48.wav	۳۲/۹۵	۶۰/۳۱	۲۰/۴۹	۵۹/۱۶	۲۹/۶۳	۴۴/۸۲
Soprano(Mozart)	۲۹/۰۳	۷۴/۶۲	۱۸/۵۶	۵۹/۴۷	۲۹/۷۴	۴۶/۰۲
Soloists(Verdi)	۲۶/۶۳	۶۴/۵۶	۳۶/۴۴	۳۵/۴۵	۳۱/۵۸	۱۸/۹۸
Track 37.wav	۳۵/۹۳	۲۸/۶۴	۴۵/۵۲	۲۳/۸۴	۳۲/۶۳	۱۲/۳۴
Electronic gong	۴۱/۶۰	۱۴/۳۲	۳۱/۷۹	۴۸/۴۳	۲۹/۵۶	۲۰/۱۹

۵-۶- ضریب موازنه

یکی از پارامترهای مهم روش مبتنی بر وجلت، ضریب موازنه، λ ، است. بهترین راه برای دست یابی به نرخ بیت‌ها و کیفیت‌های متفاوت، و کنترل عملکرد سیستم ارائه شده، تغییر این پارامتر است. در این جا قصد داریم تا نحوه‌ی تغییر SNR و نرخ بیت را برای ۴ مقدار مختلف λ بررسی کنیم. مقادیر λ در این آزمایش، 5×10^{-5} ، 9.4×10^{-4} ، 0.01، 0.1 ، هستند. سیگنال‌های مورد استفاده s1.wav، piano.wav، Soprano(Mozart).wav و track 37.wav می‌باشند. نوع کوانتیزاسیون، DPCM، است. نتایج این آزمایش را در شکل ۲-۶ و شکل ۳-۶ می‌بینید.

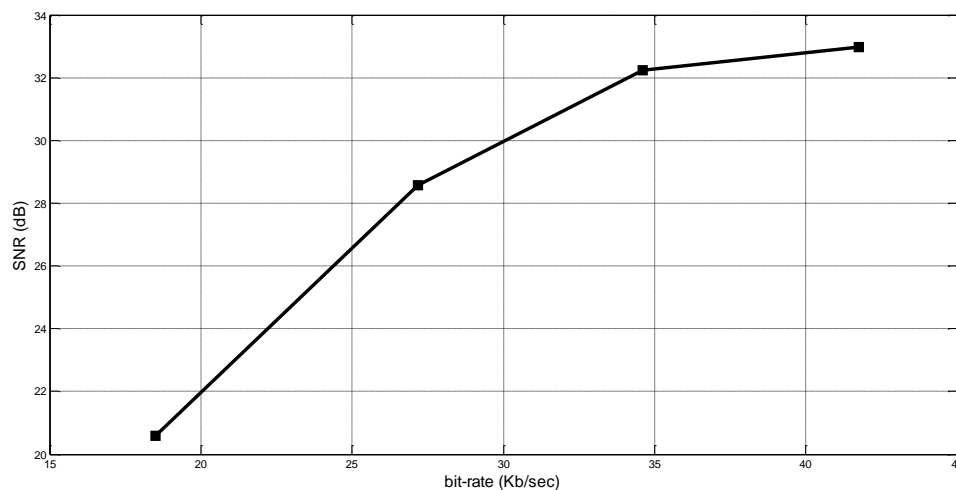


شکل ۲-۶. نتایج تغییر SNR با تغییر λ ، برای سیگنال‌های مختلف



شکل ۳-۶. نتایج تغییر نرخ بیت با تغییر λ ، برای سیگنال‌های مختلف

همان طور که از این شکل‌ها می‌بینیم، با افزایش μ ، نرخ بیت کاهش یافته و از طرف دیگر SNR سیگنال بازسازی شده نیز پایین می‌آید. وقتی ضریب موازنه خیلی به صفر نزدیک می‌شود، بهبود SNR قابل توجهی نداشته و فقط نرخ بیت افزایش می‌یابد. ممکن است یک نفر این نتیجه را بگیرد که در سیگنال‌های صحبت (مثلاً سیگنال s1.wav)، روش پیشنهادی می‌تواند در SNR حدود ۲۰، به نرخ بیت پایین‌تر از 17.12Kb/s (یعنی حدود بیش از ۴۱ برابر فشرده‌سازی) برسد. در شکل ۴-۶، میانگین نتایج این آزمایش را آورده‌ایم تا نحوه‌ی تغییرات نرخ بیت و SNR را با هم یک جا مشاهده کنیم. شکل ۴-۶، دید مناسبی از تغییرات نرخ بیت و SNR را در کنار هم نشان می‌دهد.

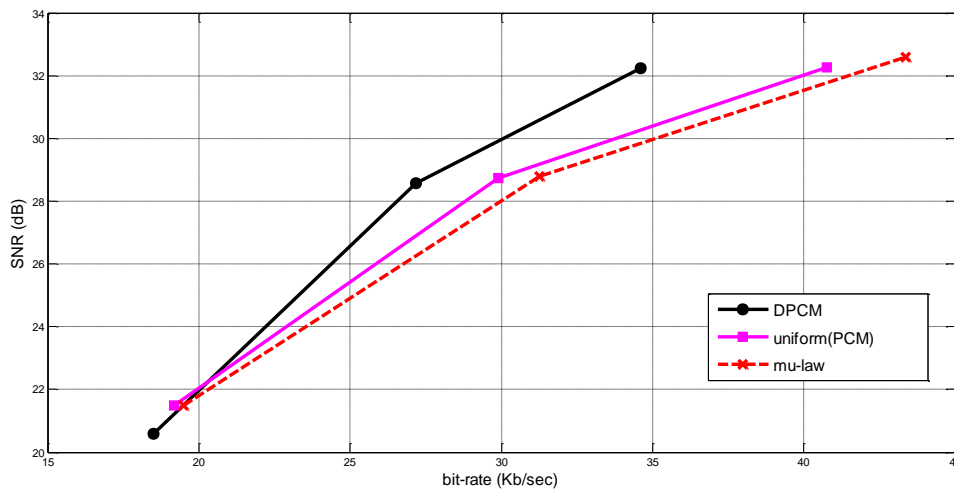


شکل ۴-۶. نحوه‌ی تغییر نرخ بیت و SNR در روش مبتنی بر وجلت (در چهار μ ی مختلف)

۴-۶- کوانتیزاسیون

در فصل ۲، سه نوع کوانتیزاسیون را معرفی کردیم که هم از نوع با حافظه و بدون حافظه، و هم از نوع یکنواخت و غیریکنواخت بودند. این سه روش عبارت بودند از: کوانتیزاسیون یکنواخت، کوانتیزاسیون غیریکنواخت به روش $\mu - law$ ، و کوانتیزاسیون با حافظه‌ی DPCM. گفته می‌شود که روش $\mu - law$ ، از آن جا که برای دامنه‌های کم، پله‌های کوتاه، و برای دامنه‌های بزرگ، پله‌های بزرگ در نظر می‌گیرد، باعث بهبود SNR نسبت به روش کوانتیزاسیون یکنواخت می‌شود، و بنابراین برای کوانتیزاسیون در حوزه‌ی زمان مناسب‌تر است. آزمایش‌های ما نشان می‌دهد که این گفته فقط

زمانی درست است که آنتروپی را در نظر نگیریم. به عبارت بهتر، هر چند سیگنال به نویز در روش $\mu - law$ نسبت به کوانتیزاسیون یکنواخت بهتر است، اما آنتروپی نمادهای به دست آمده زیاد هستند. بنابراین زمانی که بخواهیم به دنبال کوانتیزاسیون، کدگذاری آنتروپی انجام دهیم، باید به این نکته توجه داشته باشیم. در این جا، هر سه روش کوانتیزاسیون را روی مقادیر انتخاب شده (به نمایندگی از ضرایب چندجمله‌ای) روی سیگنال‌های آزمایش قبلی اعمال می‌کنیم. میانگین نتایج بدست آمده برای هر سه روش، در شکل ۵-۶ نشان داده شده است.



شکل ۵-۶. مقایسه‌ی نتایج سه روش کوانتیزاسیون

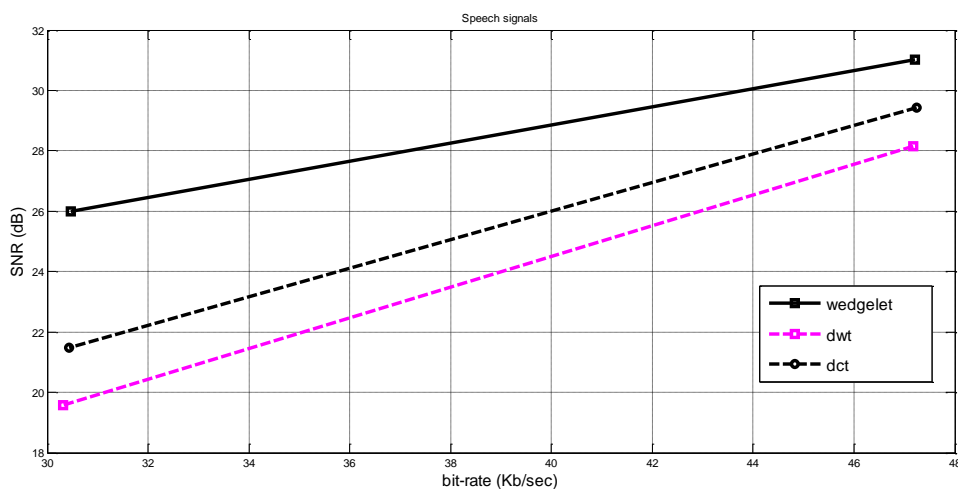
از این نمودار، به راحتی مشاهده می‌شود که روش DPCM بسیار بهتر از دو روش دیگر عمل کرده است. در نرخ بیت‌های بالا، میزان بهبود روش DPCM بیشتر شده است. همان طور که قبلاً نیز توضیح داده‌ایم دلیل این امر، همبستگی بالای بین نقاط انتخاب شده در نرخ بیت بالاست. در نرخ بیت پایین نیز، تفاوت زیادی بین آن‌ها وجود ندارد. بنابراین علاوه از فشرده‌سازی با استفاده از چندجمله‌ای‌ها، یک فشرده‌سازی مضاعف نیز، با بهره گرفتن از همبستگی بین مقادیر حاصل، انجام می‌شود.

نکته‌ی دیگر این نمودار، این است که SNR روش $\mu - law$ در تعداد بیت‌های کوانتیزاسیون برابر (و بنابراین در نرخ بیت برابر) از کوانتیزاسیون یکنواخت بیشتر است. اما تأثیر

آنتروپی بالای نمادهای حاصل از این روش، باعث شده است تا بعد از کدگذاری آنتروپی، نرخ بیت آن بیشتر از حالت یکنواخت باشد، و بنابراین نتایج کوانتیزاسیون یکنواخت، نسبت به $\mu - law$ رضایت بخش هستند.

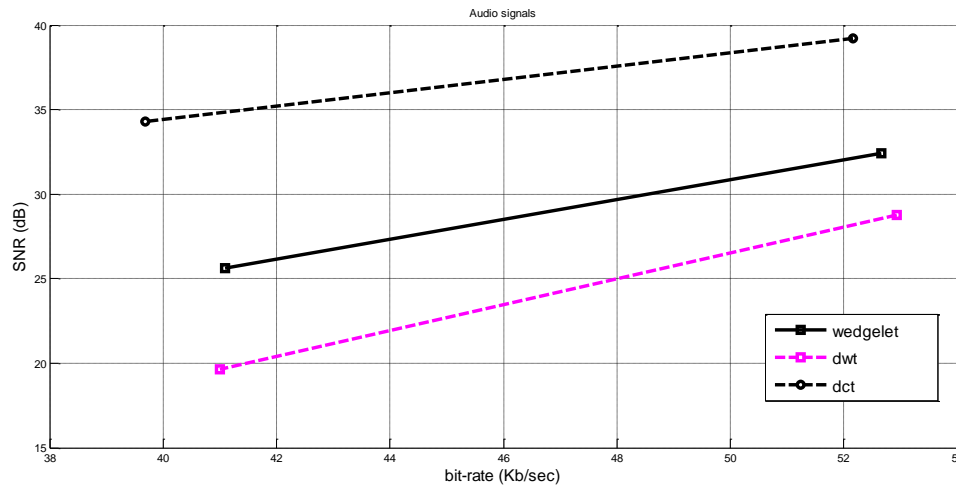
۶-۷- مقایسه

برای این که یک دیدی نسبت به عملکرد روش مبتنی بر وجلت داشته باشیم، آن را در دو نرخ بیت متفاوت و بر روی تعدادی از سیگنال‌های موجود، با روش‌های DWT و DCT مقایسه کرده‌ایم. در این جا از دو معیار SNR و MOS برای ارزیابی استفاده کرده‌ایم. با توجه با این که در روش DCT، بین سیگنال‌های صحبت و صوت، تفاوت زیادی در عملکرد این روش دیده می‌شود (جدول ۶-۲)، لذا میانگین‌گیری از نتایج بر روی تمام این سیگنال‌ها، راه درستی نیست. به همین دلیل در این قسمت، برای مقایسه، سیگنال‌های صوت و صحبت را از هم جدا کرده‌ایم. در مقایسه‌ی عملکرد روش‌ها، برای سیگنال‌های صحبت، چهار سیگنال s1.wav، s2.wav، s3.wav و s5.wav را انتخاب کرده‌ایم. برای مقایسه‌ی سیگنال‌های صوت نیز، چهار سیگنال piano.wav، saxophone.wav، track 48.wav و soloists(Verdi).wav را مورد استفاده قرار داده‌ایم. شکل ۶-۶، نتایج این آزمایش را برای سیگنال‌های صحبت، از دید SNR نشان می‌دهد.



شکل ۶-۶. نتایج مقایسه‌ی SNR روی سیگنال‌های صحبت

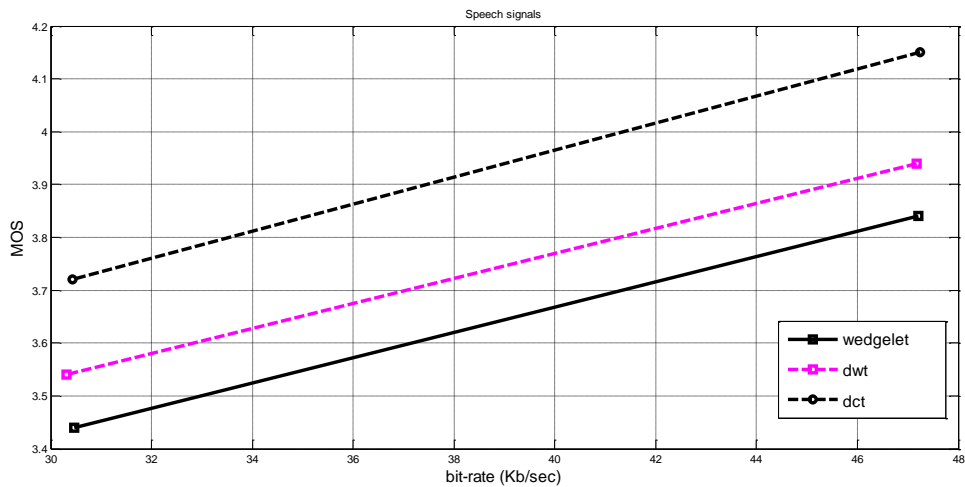
از نمودار شکل ۶-۶ به خوبی دیده می‌شود که روش پیشنهادی در مورد سیگنال‌های صحبت، بهتر از هر دو روش DWT و DCT عمل کرده است. هم‌چنین با کاهش نرخ بیت، میزان کاهش SNR در روش پیشنهادی کم است. به عبارت دیگر در نرخ بیت پایین‌تر، اختلاف نتیجه، زیادتر شده است.



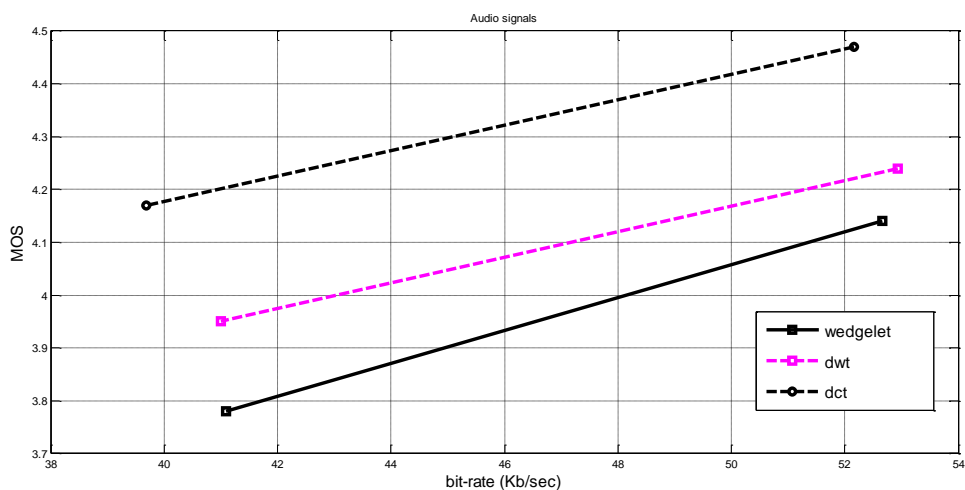
شکل ۶-۷. نتایج مقایسه‌ی SNR روی سیگنال‌های صوتی

همان‌طور که در شکل ۶-۷ دیده می‌شود، عملکرد روش مبتنی بر تبدیل DCT، در مورد سیگنال‌های صوتی از پایگاه داده، بهبود زیادی داشته است. طوری که SNR آن از روش پیشنهادی بیشتر شده است. از این‌جا نمی‌توان نتیجه گرفت که عملکرد روش پیشنهادی در مورد سیگنال‌های صوتی ضعیف‌تر شده است، بلکه باید گفت که عملکرد روش DCT بهتر شده است، و روش پیشنهادی و روش DWT، نتایج مشابهی با سیگنال‌های صحبت دارند.

شکل ۶-۸ و شکل ۶-۹ نتایج نظرسنجی MOS را روی همین سیگنال‌ها نشان می‌دهد. از روی این نمودارها مشاهده می‌شود که روش DCT بیشترین میزان MOS را دارد. البته باید به این نکته نیز توجه داشت که در نظرسنجی به روش MOS برای اعتبار نتایج، باید اذاتی که در آن نظرسنجی انجام می‌شود، دارای عایق صوتی باشد، و هم‌چنین شنوندگان نیز باید افراد ماهر و آشنا به سیگنال‌های صوتی باشند [۴۱]. ایجاد این چنین شرایط مناسب برای نظرسنجی، مشکل می‌باشد. هم‌چنین نتایج روش MOS از یک نظرسنجی به نظرسنجی دیگر، تا ۰/۴ اختلاف دارد [۴۱].



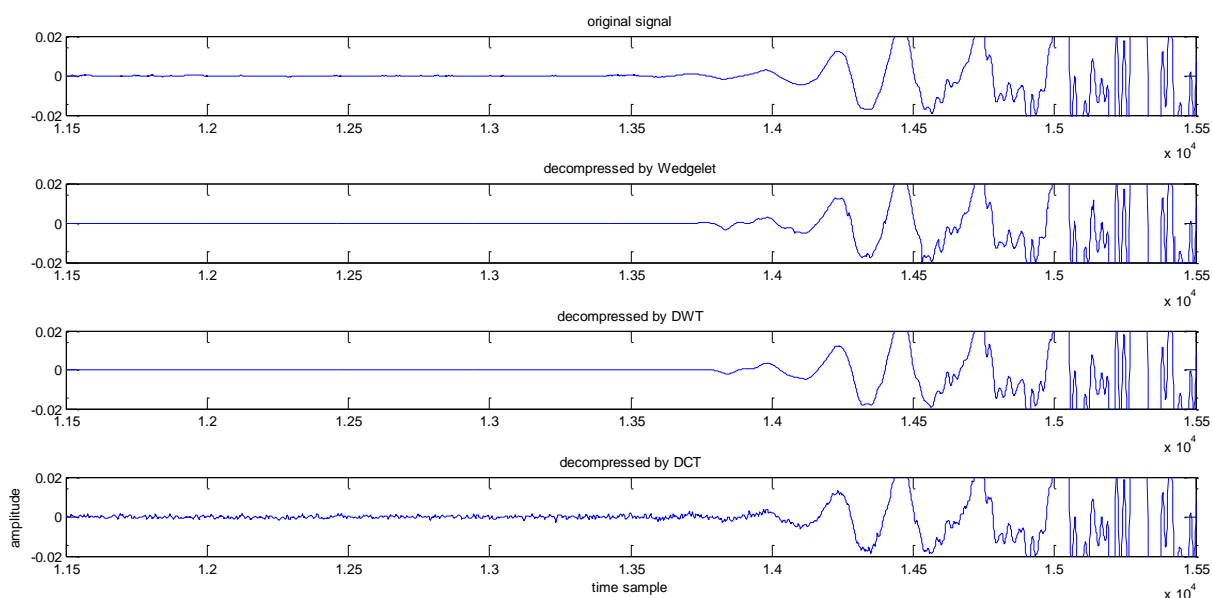
شکل ۶-۸. نتایج مقایسه‌ی MOS روی سیگنال‌های صحبت



شکل ۶-۹. نتایج مقایسه‌ی MOS روی سیگنال‌های صوتی

یکی از مواردی که در سیگنال‌های بازسازی شده از روش DCT دیده می‌شود این است که در قسمت‌های سکوت از سیگنال و یا در قسمت‌هایی با دامنه‌های کم که نزدیک به قسمت‌های با دامنه‌ی بزرگ هستند، سیگنال بازسازی شده از روش DCT، صفر نبوده بلکه یک سیگنال با دامنه‌ی کم به چشم می‌خورد. در حالی که در روش پیشنهادی و روش DWT این مورد دیده نمی‌شود و قسمت‌های سکوت، کاملاً به صورت سیگنال با مقدار صفر بازسازی می‌شوند. این مورد را می‌توانید در شکل ۶-۱۰ مشاهده کنید. دلیل این امر به سیستم‌های پایه‌ی تبدیل DCT مربوط می‌شود. چرا که توابع پایه در

تبدیل DCT، کسینوسی‌ها هستند که محلی نمی‌باشند. یعنی توابع پایه‌ی تبدیل DCT در کل بازه‌ی زمانی مقدار دارند. و بنابراین وقتی که ضرایب مربوط به تعدادی از این توابع پایه را (که دامنه‌ی کمی دارند) صفر می‌کنیم، تأثیر این توابع پایه از سیگنال حذف می‌شود و بنابراین در سیگنال بازسازی شده مقادیری غیر از صفر ایجاد می‌شود. اما در تبدیل DWT از آن‌جا که توابع پایه‌ی مربوط به این تبدیل، محلی می‌باشند، با حذف تعدادی از این توابع، فقط قسمت کوچکی از سیگنال دچار اعوجاج می‌شود. در مورد روش وجلت نیز، از آن‌جا که این روش، توانایی وفق دادن محلی تابع پایه را با سیگنال دارد، این مشکل پیش نمی‌آید. البته برای کاهش تأثیر این مورد در روش DCT می‌توان طول فریم‌ها را کوچک‌تر در نظر گرفت.



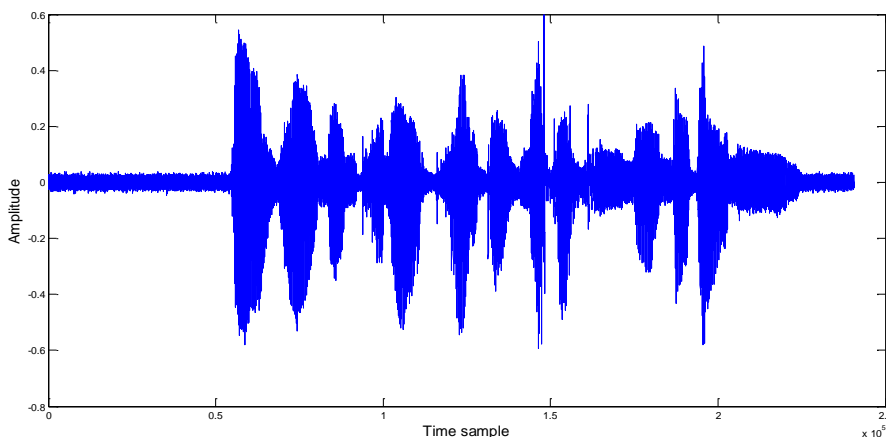
شکل ۶-۱۰. یک مثال از سیگنال‌های بازسازی شده توسط هر سه روش وجلت، DWT و DCT

۶-۸- بررسی تأثیر نویز

با توجه به این که تقریب چندجمله‌ای به روش مینیمم مربع خطا، سیگنال را به صورتی نرم‌تر^۱ تقریب می‌زند، لذا در محیط‌های نویزی، خاصیت حذف نویز از خود نشان دهد. در این قسمت قصد داریم تا با اضافه کردن نویز گوسی به تعدادی از سیگنال‌های پایگاه داده، عملکرد روش پیشنهادی را

^۱ Smooth

روی سیگنال‌های نویزی ارزیابی کنیم. برای این کار نویز گوسی با دو انحراف معیار مختلف، $\sigma = 0.01$ و $\sigma = 0.005$ به سیگنال‌های تحت آزمایش اضافه کرده‌ایم و سپس سیگنال‌ها را با هر سه روش مذکور، فشرده کرده و کدگشایی کرده‌ایم. برای ارزیابی نتایج، از دو معیار SNR و MOS استفاده کرده‌ایم. اما از آن جا که هدف ما در این قسمت بررسی توانایی حذف نویز روش پیشنهادی می‌باشد، لذا سیگنال به نویز سیگنال بازسازی شده را نسبت به سیگنال اصلی بدون نویز در نظر گرفته‌ایم. در این آزمایش از چهار سیگنال s2.wav، s5.wav، piano.wav و saxophone.wav استفاده کرده‌ایم. شکل ۶-۱۱ یک مثال از سیگنال s5.wav را که نویز گوسی با انحراف معیار $\sigma = 0.01$ به آن اضافه شده است، نشان می‌دهد. نتایج این آزمایش در جدول ۶-۳ و جدول ۶-۴ آورده شده است.



شکل ۶-۱۱. یک مثال از سیگنال نویزی شده با انحراف معیار نویز $\sigma = 0.01$

جدول ۶-۳. مقایسه‌ی نتایج SNR و MOS حاصل از سیگنال نویزی با انحراف معیار نویز $\sigma = 0.005$

سیگنال	SNR	روش پیشنهادی		روش DWT		روش DCT	
		SNR	MOS	SNR	MOS	SNR	MOS
S2.wav	۲۰/۹۱	۲۷/۳۸	۳/۸۳	۲۳/۸۷	۳/۵	۲۳/۰۹	۳/۹۳
S5.wav	۲۴/۳۵	۲۹/۳۳	۳/۷۱	۲۴/۳۱	۳/۸۲	۲۵/۴۷	۴/۰۵
Piano.wav	۱۹/۴۲	۲۳/۹۶	۳/۳۹	۲۲/۶۷	۳/۹۶	۲۲/۵۹	۳/۸۵
Saxophone.wav	۲۳/۴۸	۲۹/۹۰	۳/۷۸	۲۳/۸۲	۳/۶۴	۲۶/۱۰	۴/۱۶

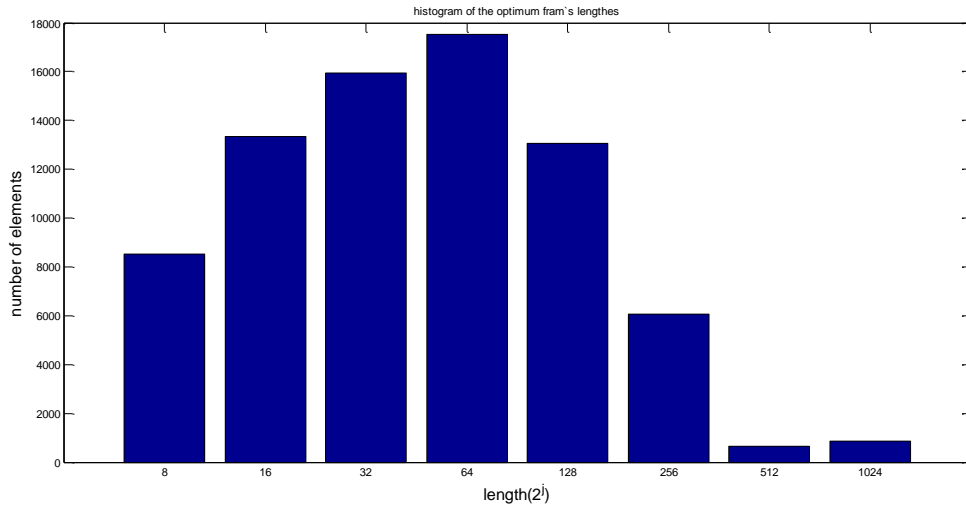
جدول ۶-۴. مقایسه‌ی نتایج SNR و MOS حاصل از سیگنال نویزی با انحراف معیار نویز $\sigma = 0.01$

سیگنال	SNR	روش پیشنهادی		روش DWT		روش DCT	
		SNR	MOS	SNR	MOS	SNR	MOS
S2.wav	۱۴/۹۹	۲۱/۱۹	۳/۱۱	۱۸/۱۷	۳/۲۲	۱۸/۵۳	۳/۵
S5.wav	۱۸/۳۶	۲۲/۵۳	۲/۹	۱۸/۳۴	۳/۲۹	۱۸/۲۱	۳/۴۶
Piano.wav	۱۳/۵۵	۲۱/۳۸	۲/۷۳	۱۷/۶۰	۳/۱۵	۱۸/۰۱	۳/۳۵
Saxophone.wav	۱۷/۵۱	۲۵/۶۹	۳/۶۴	۱۸/۵۱	۲/۹۸	۱۸/۹۳	۳/۲۷

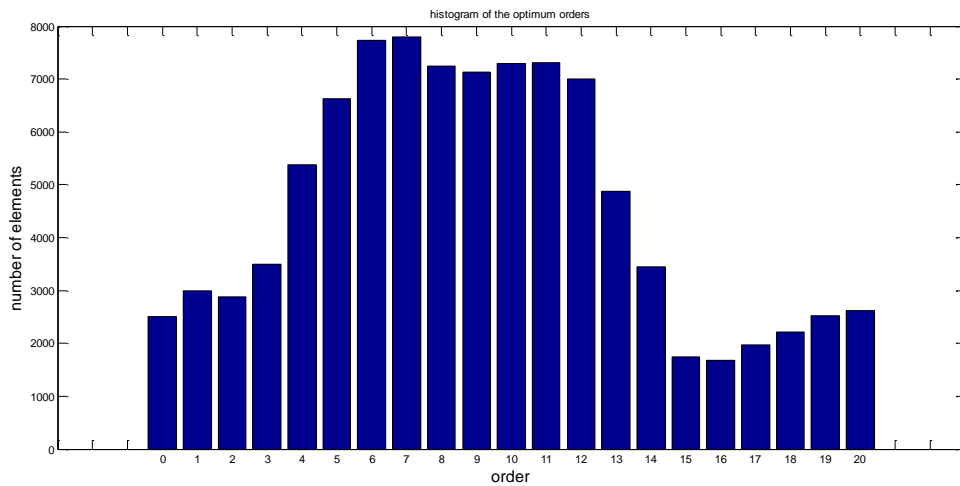
همان‌طور که از هر دو جدول بالا دیده می‌شود، از لحاظ معیار SNR روش پیشنهادی در تمامی موارد بهتر از هر دو روش دیگر عمل کرده است. از لحاظ MOS در بعضی موارد نتایج بهتری دارد. در سیگنال‌های s2.wav و saxophone.wav و در نویز کمتر، MOS مربوط به روش پیشنهادی بهتر از DWT ارزیابی شده است. هم‌چنین در سیگنال saxophone.wav و در نویز بیشتر، نتایج حاصل از نظرسنجی برای روش پیشنهادی، بهتر از دو روش دیگر ارزیابی شده است. در اکثر مواقع، روش DCT میزان MOS بیشتری دارد.

در شکل ۶-۱۲ هیستوگرام مربوط به طول فریم‌های بهینه و در شکل ۶-۱۳ نیز هیستوگرام مراتب چندجمله‌ای‌های بهینه را روی تمام سیگنال‌های پایگاه داده، آورده‌ایم. همان‌طور که دیده می‌شود در نمودار مربوط به طول فریم‌های انتخاب شده، همه‌ی طول‌ها و به تعداد مناسب و نه چندان کمی جزء طول‌های بهینه انتخاب شده‌اند که این مسأله، می‌تواند نشان از مؤثر بودن الگوریتم بهینه‌سازی درخت تجزیه باشد. هم‌چنین از روی شکل ۶-۱۳ با توجه به این که چندجمله‌ای‌های با مراتب مختلف به تعداد زیادی به عنوان توابع بهینه انتخاب شده‌اند می‌تواند دلیل بر مؤثر بودن الگوریتم بهینه‌سازی نرخ-اتلاف باشد. در این آزمایش، مراتب چندجمله‌ای‌ها را به عدد ۲۰ محدود

کرده‌ایم. دلیل این که طول پنجره‌های بزرگ، کمتر به عنوان گره‌های بهینه انتخاب شده‌اند نیز همین است.



شکل ۶-۱۲. هیستوگرام طول گره‌های بهینه



شکل ۶-۱۳. هیستوگرام مربوط به مراتب چندجمله‌ای‌های بهینه

۹-۶- ارتباط روش پیشنهادی و موجک‌ها

روش مبتنی بر وِجِلت، ارتباط نزدیکی با موجک‌ها دارد. به این شکل که روش مبتنی بر وِجِلت نیز همانند موجک‌ها یک تحلیل چند مقیاسی است و شامل توابع پایه‌ی مختلف در مقیاس هاس

مختلف می‌باشد. اما یک فرق اساسی بین آن‌ها وجود دارد. روش مبتنی بر وجلت بعضی خواص را از درخت تجزیه به ارث می‌برد، به این صورت که توانایی وفق دادن و تغییر دادن محلی تابع پایه را با سیگنال دارد. اما در موجک، توابع پایه تغییر نمی‌کنند. علاوه از آن، در روش مبتنی بر وجلت، بازسازی یک فریم یا یک نمونه از سیگنال، فقط به یک تابع پایه وابسته است، برخلاف موجک‌ها که چندین تابع پایه برای بازسازی همان قسمت یا نمونه از سیگنال، لازم هستند.

فصل هفتم

نتیجه‌گیری و پیشنهاد

۷-۱- نتیجه‌گیری

در این پایان‌نامه یک روش جدید فشرده‌سازی سیگنال‌های صوتی از نوع فشرده‌سازهای شکل‌موج ارائه شد. ایده‌ی اصلی این روش از بحث وجلت‌ها که یکی از ابزارهای پردازش تصویر می‌باشد، گرفته شده است.

تعمیم این روش فشرده‌سازی روی سیگنال‌های صوتی، به ارائه‌ی یک روش فشرده‌سازی از نوع فشرده‌سازهای شکل‌موج انجامید. توابع پایه‌ی مربوط به وجلت را توابع چندجمله‌ای از مراتب مختلف انتخاب کرده‌ایم. کوانتیزاسیون ضرایب مربوط به این چندجمله‌ای‌ها به دلیل محدودده‌ی دینامیکی بالای این ضرایب مشکل‌ساز است. این مسأله باعث شد تا مقادیر مربوط به سیگنال کد شده را از مقادیر حوزه‌ی زمان سیگنال انتخاب کنیم. لذا برای کوانتیزاسیون این مقادیر سه روش کوانتیزاسیون PCM، DPCM و μ -law را امتحان کردیم که نتایج روش DPCM نسبت به دو روش دیگر بهتر شد. هم‌چنین عملکرد سیستم را در محیط‌های نویزی نیز مورد بررسی قرار دادیم.

نرخ بیت و کیفیت سیگنال در روش پیشنهادی، قابل کنترل و تنظیم می‌باشد. موازنه‌ی بین نرخ بیت و کیفیت سیگنال، با استفاده از تابع هزینه‌ی لاگرانژ برقرار شد. در این تابع، معیار اتلاف را براساس ماکزیمم کردن SNR در نظر گرفتیم. نتایج حاصل به ما نشان دادند که SNR سیگنال بازسازی شده در اکثر سیگنال‌ها، بالاتر از روش‌های بررسی شده‌ی موجود است.

یکی دیگر از ویژگی‌های روش پیشنهادی، سرعت بالا و دقت خوب آن است. به عبارت دیگر، از آن جا که تمامی طول‌های ممکن را برای فریم‌ها در نظر نمی‌گیریم، و پردازش، فقط بر روی فریم‌هایی با طول‌های خاص (که متناظر با مقیاس خاصی هستند) انجام می‌شود، روش پیشنهادی سرعت بالایی دارد. از طرف دیگر دقت تقریب بسیار خوبی هم دارد زیرا این روش، مبتنی بر تحلیل چند مقیاسی بوده و فریم‌ها در سطوح تجزیه‌ی مختلف با یکدیگر مقایسه می‌شوند. یعنی یک فریم را با مجموع تقریب‌های زیرفریم‌های کوچک‌تر (فریم‌های فرزند) مقایسه و بهینه‌سازی می‌کنیم.

در روش پیشنهادی، قسمت کدگشایی الگوریتم، سریع‌تر از قسمت کدگذاری می‌باشد. این به دلیل الگوریتم‌های بهینه‌سازی در قسمت کدگذاری است. لذا این روش می‌تواند در کاربردهای برون‌خط^۱ مورد استفاده قرار بگیرد.

یکی دیگر از ویژگی‌های روش پیشنهادی این است که اگر قسمتی از سیگنال، ثابت باشد (مانند قسمت‌های سکوت) روش پیشنهادی، این قسمت را با یک چندجمله‌ای از مرتبه‌ی صفر تقریب می‌زند و بنابراین تعداد بیت‌های بسیار کمی به این قسمت اختصاص می‌یابد. به همین دلیل نیازی به استفاده از کدگذاری دوره‌ی تداوم (که در روش‌های DCT و DWT استفاده می‌شود) نداریم. علاوه بر آن اگر سیگنال در قسمت سکوت، تغییرات بسیار اندکی به واسطه‌ی نویز نیز داشته باشد، این قسمت‌ها نادیده گرفته شده و کل قسمت سکوت، با یک چندجمله‌ای مرتبه‌ی صفر، نمایندگی می‌شود.

در این پایان‌نامه، یک ایده برای فشرده‌سازی مبتنی بر نواحی مطلوب نیز ارائه شد که از واریانس فریم‌ها به عنوان ویژگی استفاده می‌کند تا قسمت‌های صحبت را با کیفیت بهتر و قسمت‌های سکوت را با نرخ بیت پایین‌تر فشرده کند.

۷-۲- پیشنهاد برای ادامه‌ی کار

یکی از ویژگی‌های روش ارائه شده این است که پارامتر اتلاف را در تابع هزینه، می‌توان براساس معیارهای اندازه‌گیری کیفیت مختلف تنظیم کرد. در این پایان‌نامه دیدیم که اگر معیار اتلاف را می‌نیمم انرژی اختلاف دو سیگنال در گره‌های درخت تجزیه در نظر بگیریم، در نهایت SNR مربوط به کل سیگنال بازسازی شده، ماکزیمم می‌شود. نتایج فصل ششم نیز نشان دهنده‌ی این بودند که SNR روش پیشنهادی قابل قبول و امیدبخش است.

^۱ Off-Line

یک معیار دیگر اندازه‌گیری کیفیت، سیگنال به نویز قطعه‌ای^۱ است. به طریق مشابه اگر در تک تک فریم‌ها، معیار کیفیت را ماکزیمم کردن SNR در نظر بگیریم، در نهایت، SegSNR مربوط به سیگنال کلی ماکزیمم می‌شود. نتایج پیاده‌سازی‌های ما هم این نتیجه را تأیید می‌کند. بنابراین مشاهده می‌کنیم که پارامتر اتلاف را در تابع هزینه، روی هر معیار کیفیت که تنظیم می‌کنیم، از دید معیار مذکور به مقادیر قابل قبولی می‌رسیم. با این توصیف، پیشنهاد ما برای ادامه‌ی این کار، این است که معیار اتلاف را براساس مدل‌های صوتی-ادراکی^۲ در نظر بگیریم. به نظر می‌رسد در این صورت نتایج حاصل از کیفیت شنیداری سیگنال با توجه به الگوریتم بهینه‌سازی نرخ-اتلاف بهبود قابل توجهی داشته باشد.

¹ Segmental SNR

² Psychoacoustic Models

پایگاه داده

سیگنال	توصیف	طول (ثانیه)
s1.wav	صحبت (زن)	۴/۲۷
s2.wav	صحبت (زن)	۷/۴۳
s3.wav	صحبت (زن)	۵/۵۳
s4.wav	صحبت (مرد)	۵/۰۲
s5.wav	صحبت (مرد)	۵/۴۳
s6.wav	صحبت (مرد)	۵/۷۶
Violon.wav	موسیقی ویولن	۱۷/۱۸
Piano(Schubert).wav	موسیقی پیانو	۱۴/۴۹
Castanet.wav	موسیقی فرکانس بالا	۸/۵۹
Saxophone.wav	موسیقی ساکسوفون (نوعی آلت موسیقی بادی) (موسیقی فرکانس پایین)	۱۳/۵۶
Track 48.wav	آواز چند نفره	۱۲/۴۵
Soprano(mozart).wav	ترانه (موسیقی همراه با آواز زن)	۱۱/۳۳
Soloists(Verdi).wav	موسیقی همراه با آواز مرد	۱۳/۴۷
Track 37.wav	نوعی صدای زنگ مانند	۱۳/۵۶
Electronic gong.wav	ناقوس الکترونیکی (400Hz)	۵/۶۷

مراجع

- [1] F. Wylie, "Digital audio data compression", *Electronics and Communication Engineering Journal*, vol. 7, pp. 5–10, Feb. 1995.
- [2] H. M. D. Kabir¹, S. B. Alam, Md. I. Azam, R. Ahmed, "A Theory of Loss-less Compression of High Quality Speech Signals with Comparison", *UKSim Fourth European Modelling Symposium on Computer Modelling and Simulation*, DOI 10.1109/EMS.2010.33, 2010 IEEE
- [3] C.E. Shannon, "A mathematical theory of communication", *Bell Sys. Tech. J.*, vol. 27, pp. 379–423, pp. 623–656, July/Oct., 1948.
- [4] Abdul Mawla M. A. najih, A. R. bin Ramli, V. Prakash, and Syed A. R., "Speech Compression Using Discreet Wavelet Transform", *Proceedings of 4th National Conf. on Telecommunication Technology IEEE*, pp.1-4, March 2003.
- [5] H. M. Magboun, N. Ali, M. A. Osman, S. A. Alfandi, "Multimedia Speech Compression Techniques", *3rd IEEE International Conference on Computer Science and Information (ICCSIT)*, pp. 498- 502, 2002.
- [6] A. Spanias, T. Painter, V. Atti, "Audio Signal Processing and Coding", *Published by John Wiley & Sons, Inc.*, Hoboken, New Jersey. 2007
- [7] A. Gersho and R. M. Gray, "Vector Quantization and Signal Compression", *Massachusetts: Kluwer Academic Publisher*, 1991
- [8] R. Schafer and J. Markel, "Speech Analysis", *IEEE Press*, New York, 1979.
- [9] N. Jayant and P. Noll, "Digital Coding of Waveforms Principles and Applications to Speech and Video", *Prentice-Hall*, Englewood Cliffs, NJ, 1984.
- [10] Prof. Dr. Hans-Georg Stark, "Wavelets and Signal Processing, An Application-Based Introduction", ISBN 3-540-23433-0 Springer Berlin Heidelberg New York 2005
- [11] T. Cover and J. Thomas, "Elements of Information Theory", *Wiley-Interscience*, New York, Aug. 1991.

- [12]J. Ziv and A. Lempel, “A universal algorithm for sequential data compression”, *IEEE Trans. Inf. Theory*, vol. IT-23, no. 3, pp. 337–343, May 1977.
- [13]I.H. Witten, R.M. Neal, and J.G. Cleary, “Arithmetic Coding for data compression”, *Comm. ACM*, vol. 30, no. 6, pp. 520–540, June 1987.
- [14]M. Misiti, Y. Misiti, G. Oppenheim and J. Poggi, “Matlab Wavelet Tool Box”, *The Math Works Inc.*, 2000.
- [15]Martin Vetterli, Jelena Kovacevic, “Wavelets and Subband Coding”, published by Prentice Hall PTR, Englewood Cliffs, New Jersey. 2007.
- [16]M. Kahrs, K. Brandenburg, “Applications of Digital Signal Processing to Audio and Acoustics”, 2002 *Kluwer Academic Publishers*:New York, Boston, Dordrecht, London, Moscow
- [17]R. J. McAulay and T. F. Quatieri, “Speech Analysis/Synthesis Based on a Sinusoidal Representation”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. assp-34, no. 4, august 1986
- [18]E. B. George and M.J. T. Smith, “Analysis by synthesis/overlap add sinusoidal modeling applied to the analysis and synthesis of musical tones”, *Journal of the Audio Engineering Society*, vol.40, no. 6, pp. 1987 1990,1989
- [19]E. B. George and M.J. T. Smith, “Speech Analysis/synthesis and modification using an Analysis by synthesis overlap-add sinusoidal model”, *IEEE transaction on speech and audio processing*, Vol. 5, No. 5,pp. 389-406, Sep. 1997
- [20]S. G. Mallat, and Zhifeng Zhang, “Matching Pursuit With Time-Frequency Dictionaries”, *IEEE transactions on signal processing*, Vol. 41, No. 12, pp. 3397-3415, Dec. 1993.
- [21]L. B. Almeida and F. M. Silva, “Variable-frequency synthesis: An improved harmonic coding scheme”, in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, San Diego, CA, 1984, p. 27.5.1.
- [22]R. J. McAulay and T. F. Quatieri, “Magnitude-only reconstruction using a sinusoidal speech model,” in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, San Diego, CA, 1984, p. 27.6.1.
- [23]R. J. McAulay and T. F. Quatieri, “Mid-rate coding based on a sinusoidal representation of speech”, in *Proc. ICASSP-8*, (Tampa, FL, Apr. 1985), pp. 945.

- [24] D. Kirovski, and Z. Landau, “Generalized Lempel–Ziv Compression for Audio”, *IEEE Transactions on Audio, Speech, and Language processing*, vol. 15, no. 2, february 2007
- [25] H. Malvar, “A modulated complex lapped transform and its application to audio processing”, in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1999, pp. 1421–1424.
- [26] M. Hans, “Optimization of digital audio for Internet transmission”, Ph.D. dissertation, School Elect. Comp. Eng., Georgia Inst. Technol., Atlanta, 1998.
- [27] *Sound Quality Assessment Material Recordings for subjective Tests*. Technical Centre of the European Broadcasting Union, 1988
- [28] M. Purat, T. Liebchen, and P. Noll, “Lossless transform coding of audio signals”, in *Proc. 102nd AES Conv.*, Munich, Germany, 1997, preprint 4414.
- [29] Craven, P.; Gerzon, M. “*Lossless Coding for Audio Discs*”, *AES journal*, September 1996.
- [30] T. Robinson, “SHORTEN: Simple lossless and near-lossless waveform compression”, Cambridge Univ. Eng. Dept., Cambridge, UK, Tech. Rep. 156, 1994.
- [31] R. E. Walpole, R. H. Myers, “Probability & Statistics for Engineers and Scientists”, 2nd ed.
- [32] I. Daubechies, “The wavelet transform. Time–frequency localization and signal analysis”, *IEEE Trans. Inform. Theor.* 36 (1990) 961–1005.
- [33] E. Magosso, M. Ursino, A. Zaniboni, E. Gardella, “A wavelet-based energetic approach for the analysis of biomedical signals: Application to the electroencephalogram and electro-oculogram”, *Elsevier Inc. applied Mathematics and Computation* 207 (2009) 42–62
- [34] R.M. Rao, A.S. Bopardikar, “Wavelet Transforms: Introduction to Theory and Applications”, Addison Wesley, 1998.
- [35] S. G. Mallat, “A Wavelet Tour of Signal Processing”, 3 th edition, Elsevier Inc., 1999.

- [36] S. M Joseph, B. Anto P, “speech compression using wavelet transform”, *IEEE-International Conference on Recent Trends in Information Technology*, MIT, Anna University, Chennai. June 3-5, 2011
- [37] E. B.Fgee, W. J. Phillips and W. Robertson, “Comparing Audio Compression Using Wavelet with Other Audio Compression Schemes”, *IEEE Transaction Electrical and Computer Engineering* IEEE, Vol.2, pp.698-701, Aug 2002.
- [38] D. Sinha and A. Tewfik. “Low Bit Rate Transparent Audio Compression using Adapted Wavelets”, *IEEE Trans. ASSP*, Vol. 41, No. 12, December 1993.
- [39] Saeed V. Vaseghi “Multimedia Signal Processing: Theory and Applications in Speech, Music and Communications”, 2007 John Wiley & Sons Ltd, West Sussex PO19 8SQ, England
- [40] N.S. Jayant and P. Noll, “Digital Coding of Waveforms - Principles and Applications to Speech and Video”, Prentice-Hall, Englewood Cliffs (NJ) (1986).
- [41] A. S. Spanias, “Speech coding: A tutorial review”, *Proceedings of the IEEE*, vol. 82, No. 10, pp. 1541-1582, October 1994.
- [42] S. Nagaswamy, “Comparison of CELP speech coder with a wavelet method”, *A Thesis for the degree of Master of Science*, University of Kentucky, 2005
- [43] Averbuch, Amir Bobrovsky, B. Sheinin, V., “Speech compression using wavelet packet and vector quantizer with 8-msec delay”, *Proc. SPIE* Vol. 2569, p. 320-332.
- [44] A. S. Spanias, “A Hybrid Transform Method for Speech Analysis and Synthesis”, *Signal Processing*, Vol. 24, pp. 217-229, Aug. 1991.
- [45] H. kaur, R. kaur, “Speech compression and decompression using DWT and DCT”, *Int. Journal of Computer Technology & Applications(IJCTA)*, Vol 3 (4), 1501-1503, July-August 2012
- [46] G. Rajesh, A. Kumar, K. Ranjeet, “Speech Compression using Different Transform Techniques”, in *IEEE Int. Conf. on Computer & Communication Technology.*, 2011
- [47] M. A. Osman, Nasser Al, H. M. Magboub, S.A. Alfandi, “Speech compression using LPC and wavelet”, in *IEEE 2nd International Conference on Computer Engineering and Technology*, 2010

- [48] R. Kastantin, D. Stefanoiu, G. Feng, N. Martin and M. Mrayati, "Optimal Wavelets for High Quality Speech Coding", in *Proc. Of EUSIPCO-94, European Association for Signal Processing*, Edinburgh, Scotland, 1994, pp. 399-402.
- [49] Dr.V.Radha, Vimala.C, M.Krishnaveni, "Comparative Analysis of Compression Techniques for Tamil Speech Datasets", in *IEEE-International Conference on Recent Trends in Information Technology*, June 3-5, 2011
- [50] J.I. Agbinya, "Discrete Wavelet Transform Techniques in Speech Processing", in *IEEE Tencon Digital Signal Processing Applications Proceedings*, New York, NY, 1996, pp 514-519.
- [51] S. M. Joseph, F. Shah A, B. Anto P, "Spoken Digit Compression: A Comparative Study between Discrete Wavelet Transforms and Linear Predictive Coding", *International Journal of Computer Applications* (0975 – 8887) Vol. 6, No.6, September 2010
- [52] Jalal Karam, "End Point Detection for Wavelet Based Speech Compression", *World Academy of Science, Engineering and Technology* 13, 2008
- [53] N. Ahmed and K. Rao, "Orthogonal Transforms for Digital Signal Processing", *New York, Springer-Verlag*, 1975
- [54] D. L. Donoho, "Wedgelets: Nearly-Minimax Estimation of Edges", Stanford University and U.C. Berkeley, August, 1997
- [55] Romberg J., Wakin M., Baraniuk R. "Approximation and Compression of Piecewise Smooth Images Using a Wavelet/Wedgelet Geometric Model", *IEEE International Conference on Image Processing*, USA, September.
- [56] Wakin M., Romberg J., Choi H., Baraniuk R, "Geometric Tools for Image Compression", *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, USA, November. 2003
- [57] D.L. Donoho, Huo X, "Beamlets and Multiscale Image Analysis", *Lecture Notes in Computational Science and Engineering, Multiscale and Multiresolution Methods*, Springer. 2002
- [58] A. Lisowska, "Extended Wedgelets — Geometrical Wavelets in Efficient Image Coding", Institute of Mathematics, University of Silesia, Machine GRAPHICS & VISION

- [59] Prieto M.S., Allen A.R, “A Similarity Metric for Edge Images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10), 1265–1273.
- [60] S. Dusan, J.L. Flanagan, A. Karve, M. Balaraman, “Speech Compression by Polynomial Approximation”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 2, February 2007
- [61] S. A. Okunuga, “numerical analysis”, *national open university of nigeria*
- [62] Zoran Peri, Jelena Nikoli “An adaptive waveform coding algorithm and its application in speech coding”, *Elsevier Inc. Digital Signal Processing* 22 (2012) 199–209, doi:10.1016
- [63] J. L. Buchanan and P. R. Turner, “Numerical Methods and Analysis”, New York: McGraw-Hill, 1992.
- [64] R.A. Blahut, “Principles and Practice of Information Theory”, Addison-Wesley Publishing Company, Reading, MA, 1988.
- [65] P. Zolfaghari and T. Robinson, “Speech coding using mixture of Gaussians polynomial model”, in *Proc. EUROSPEECH*, 1999, pp. 1495–1498.
- [66] A.R. Madane, Z. Shah, R. Shah, S. Thakur, “Speech Compression Using Linear Predictive Coding”, *Proceeding of the international workshop on machine intelligence research*. 2009 MIR Labs.
- [67] D. Ying, Y. Yan, J. Dang, F. K. Soong, “Voice Activity Detection Based on an Unsupervised Learning Framework”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, November 2011

Abstract: Due to some considerations such as daily increasing amount of audio data, bandwidth limitation, consumed memory, need to have acceptable audio quality, and better using of available resources, our need for audio coding methods is inevitable.

Wedgelets is one of the successful tools for coding and noise reduction of images. The wedgelet decomposition consists of basis functions at multiple scales in the same way as the wavelets. Here, we employ and investigate the coding performance of extended wedgelets on audio data.

In this thesis, first, we have a review of image coding based on wedgelets and extended wedgelets. Then, we have discussed the problems of applying this method on audio signals and presented solutions for them. We used polynomial functions as the basis functions of the wedgelets. Since the coefficients we obtained are from the temporal space, and due to the correlation between these coefficients, three different methods of quantization are examined. In conjunction with our method, a Region of Interest (ROI) coding is presented to differentiate between voice and silence segments in speech signals. Performance of the proposed system is also evaluated in noisy environments. One of the advantages of the proposed method is that the compression ratio and the quality of the decompressed signal can be controlled.

For evaluating, the proposed method is compared with two transform-based methods including DWT and DCT. The results are compared using both objective and subjective measures, SNR and MOS. The results show that in terms of SNR, proposed method for speech signals, in average, has better result than DWT and DCT by 19.40% and 11.90% respectively. Also for audio signals, wedgelet based audio coding have better results over DWT by 19.97%. The study leads us to conclude that wedgelet based coding can be an efficient model for coding of different types of audio and speech signals.

Keywords: *Audio compression, wedgelets, encoding, decoding, ROI coding, pattern matching.*



Audio Coding Based on the Pattern Matching Technique

Thesis

Submitted in Partial Fulfillment of the
Requirements for the Degree of Master of Science (M. Sc.)
In Electronic Engineering

Department of Electrical Engineering and Robotics
Shahrood University of Technology

By:

Amin Ghanbarzadeh

Supervisors:

Dr. Ali Soleimani

Dr. Hadi Grailu

Advisor:

Dr. Hossein Marvi

February 2013