

به نام خدا



دانشگاه صنعتی شریف
دانشکده مهندسی برق

پروژه درس سیگنال‌ها و سیستم‌ها
گروه دکتر بهروزی

عنوان:

فشرده‌سازی صوت

اردیبهشت ۱۴۰۱

در این پروژه فشرده‌سازی صوت را به عنوان کاربرد مهمی از MDCT^۱ بررسی می‌کنیم.

۱ مقدمه

همانطور که می‌دانید فایل‌های صوتی را می‌توان با فرمت‌های متنوعی همچون MP3^۲، AAC^۳ و WMA^۴ فشرده ساخت. وجه اشتراک این فرمت‌ها در آن است که بخش اصلی فشرده‌سازی بر مبنای گونه‌ای از تبدیل DCT^۵ انجام می‌شود که MDCT نام دارد. Audio codec سیستمی است که از آن برای کدگذاری/کدبرداری فایل‌های صوتی استفاده می‌شود. در این پروژه می‌خواهیم یک طرح پایه‌ای برای Audio codec را بر مبنای MDCT پیاده‌سازی کنیم و به بررسی کیفیت صوت کدشده بپردازیم. شما در این درس با تحلیل حوزه فرکانس سیگنال‌ها آشنا شدید و تبدیل‌های مختلفی را فراگرفتید. یکی از تبدیل‌های حوزه زمان به حوزه فرکانس، تبدیل کسینوسی گسسته (DCT) است که پایه اصلی فشرده‌سازی بسیاری از دادگان محسوب می‌شود. MDCT با تغییری در DCT حاصل شده است تا برای فشرده‌سازی صوت مناسب باشد.

۲ توضیح الگوریتم

سیگنال صوت به بلوک‌هایی به طول n تقسیم می‌شود و در ادامه تبدیل MDCT اعمال می‌گردد. در الگوریتم‌های فشرده‌سازی صوت، MDCT را بر روی بردارهایی از داده اعمال می‌کنند که همپوشانی دارند. دلیل این کار پرهیز از پدیدار آمدن مصنوعات است که منجر به خطاهایی متناوب می‌شوند. گوش انسان بیشتر از چشم او به خطاهایی متناوب حساسیت دارد و به شنیدن آن تمرکز خواهد کرد. برداری به طول $2n$ که دربردارنده دو بلوک پیایی است در ورودی تبدیل MDCT در نظر گرفته می‌شود. حاصل تبدیل برداری به طول n خواهد بود و از خاصیت همپوشانی می‌توان برای بازسازی بخش مشترک دو بلوک پیایی استفاده کرد. اولین و آخرین بلوک قابل بازیابی نخواهند بود، بنابراین لازم است که به شکلی مناسب از Padding استفاده شود.

پس از اجرای تبدیل MDCT به فشرده‌سازی سیگنال می‌پردازیم. ایده اصلی آن است که می‌توانیم برخی از مولفه‌های فرکانسی را با دقت بیشتری نگه داریم و تأکید کمتری برای برخی دیگر داشته باشیم.

¹Modified DCT (MDCT)

²MPEG-1 Audio Layer 3 (MP3)

³Advanced Audio Coding (AAC)

⁴Windows Media Audio (WMA)

⁵Discrete Cosine Transform (DCT)

برای فشرده‌سازی سیگنال صوت می‌توان حاصل تبدیل MDCT را به b بیت کوانتیزه کرد که منجر به فشرده‌سازی با اتلاف و کاهش فضای لازم برای ذخیره‌سازی می‌شود. برای بازسازی سیگنال صوت کارهایی در جهت معکوس انجام می‌شوند.

برای بهبود کیفیت سیگنال بازسازی شده می‌توان از window function استفاده کرد که سیگنال ورودی x را تغییر مقیاس می‌دهد به نحوی که در ابتدا و انتها پنجره مقادیری نزدیک به صفر داشته باشد. سیگنال x دقیقاً متناوب نیست و این رویکرد تا حدی اثر آن را جبران می‌سازد. یک راهکار می‌تواند استفاده از تابع h برای پنجره‌ای به طول $2n$ باشد که مولفه i آن در رابطه ۱ تعریف شده است. از آن قبل از اعمال تبدیل MDCT و بعد از تبدیل وارون استفاده می‌شود.

$$h_i = \sqrt{2} \sin \left(\frac{(i - \frac{1}{2}) \pi}{2n} \right) \quad (1)$$

۳ خواسته‌های پروژه

۱. در مورد تبدیل‌های DCT و MDCT تحقیق کنید و تفاوت‌ها و شباهت‌های آنها را با تبدیل فوریه بررسی کنید.

۲. بر اساس توضیحات گفته شده یک Audio codec ساده را پیاده‌سازی کنید. می‌توانید از بلوک‌هایی به طول 32 یا 64 استفاده کنید. از ضرایب کوانتیزه شده سیگنال صوت را بازسازی کنید. این الگوریتم را بر روی چند pure tone اعمال کنید و کیفیت صوت بازسازی شده را بصورت کیفی و کمی بررسی کنید. معیار RMSE را در نظر داشته باشید. در مثالهایی سیگنال‌های اصلی و بازسازی شده را در حوزه زمان و فرکانس مقایسه کنید. نمودار RMSE بر حسب فرکانس tone را رسم کنید. تأثیر تغییر تعداد بیت‌های کوانتیزاسیون را بررسی کنید.

۳. در مورد window function ها مطالعه کنید و بررسی کنید که چرا اعمال این توابع می‌تواند باعث کاهش خطا شود. تأثیر استفاده از تابع معرفی شده در رابطه ۱ را بررسی کنید. در این حالت هم نمودار RMSE بر حسب فرکانس tone را رسم کنید. چه تغییری کرده است؟

۴. یک فایل با فرمت WAV را در نظر بگیرید. (به اختصار در مورد این فرمت تحقیق کنید) اگر صوت انتخاب شده بصورت stereo است، هر یک از کانالها را جداگانه پردازش کنید. الگوریتم فشرده‌سازی را اجرا سازید و کیفیت صوت بازسازی شده را به صورت کیفی و کمی بررسی کنید. در جدولی معیار RMSE را برای مقادیر متفاوتی از تعداد بیت‌های کوانتیزاسیون و با فرض حضور یا عدم حضور window function مقایسه کنید. چه نتیجه‌ای می‌گیرید؟