

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده برق و رباتیک

گروه الکترونیک

موضوع:

طراحی و پیاده سازی یک سیستم کدکننده/کدگشا برای فشرده سازی دنباله‌های ویدئویی ورزشی با تاکید بر

نواحی مطلوب

رویا سلطانی

استاد راهنما:

دکتر هادی گرایلو

استاد مشاور:

دکتر علیرضا احمدی فرد

پایان نامه جهت اخذ درجه کارشناسی ارشد

بهمن ۹۱



مدیریت تحصیلات تکمیلی
فرم شماره (۶)

بسمه تعالی

شماره: ۱۰۳۲/آ.ت.ب
تاریخ: ۹۱/۱۱/۱۴
ویرایش: -----

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:

رؤیا سلطانی رشته: برق گرایش: الکترونیک

تحت عنوان: طراحی و پیاده سازی یک سیستم کد کننده/کد گشا برای فشرده سازی دنباله های ویدئویی ورزشی با تأکید بر نواحی مطلوب

که در تاریخ ۱۳۹۱/۱۱/۱۴ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح زیر است:

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

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

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

۵- نمره کمتر از ۱۴ غیر قابل قبول

امضاء	مرتبه علمی	نام و نام خانوادگی	عضو هیأت داوران
	استادیار	هادی بگریلو	۱- استاد راهنما
	استادیار	مهدی امیرمؤید	۲- استاد مشاور
	استادیار	سنان نامع	۳- نماینده شورای تحصیلات تکمیلی
	رتبه دار	سعید مومن	۴- استاد ممتحن
	استادیار	علی مومن	۵- استاد ممتحن

با سپاس از دو وجود مقدس

آنانکه ناتوان شدند تا ما به توانایی برسیم...

موهایشان سپید شد تا ما رو سفید شویم...

و عاشقانه سوختند تا گرمابخش وجود ما و روشنگر راهمان باشند...

تقدیم به پدر عزیزم

و مادر فداکارم

و تقدیم به :

برادرانم

که همیشه همراهم بودند

به پاس تعبیر عظیم و انسانی‌اش از کلمه‌ی ایثار و از خود گذشتگی...

و به پاس محبت‌های بی دریغش که هرگز فروکش نمی‌کند...

تقدیم به :

استاد راهنمای بزرگوارم جناب آقای دکتر هادی گرایلو

بر خود لازم میدانم تا بدینوسیله از زحمات بی‌دریغ و راهنمایی‌های ارزشمند استاد راهنمایم جناب آقای دکتر گرایلو و استاد مشاورم جناب آقای دکتر احمدی فرد در راستای انجام این پروژه تشکر و قدردانی نمایم.

تعهد نامه

اینجانب *دکتر سید علی حسینی* دانشجوی دوره کارشناسی ارشد رشته *مهندسی برق* دانشکده مهندسی برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان نامه با عنوان :

طراحی و پیاده سازی سیستم کنترل کننده / کدنگ برای کنترل موتور دینامیکی و الکترونی
تحت راهنمایی آقای دکتر *دکتر سید علی حسینی* 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: ۹۱/۱۱/۱۴

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.

چکیده

امروزه در سازمانهای دولتی/غیر دولتی بسیاری در اقصی نقاط دنیا، داده‌های ویدیویی بسیاری به صورت روزافزون در حال تولید، ذخیره، و ارسال است. نگهداری یا ذخیره‌ی چنین حجم انبوهی از دنباله‌های ویدیویی مستلزم استفاده از حافظه‌های با حجم بسیار زیاد است که چندان مقرون به صرفه نبوده و علاوه بر این حجم فیزیکی نسبتاً زیادی اشغال می‌شود که منجر به مشکل شدن جابجایی و دسترسی به داده‌های مذکور می‌شود. یک راه برای غلبه نسبی بر مشکل نگهداری یا ذخیره‌ی دنباله‌های ویدیویی استفاده از روشهای موثر فشرده‌سازی ویدیو می‌باشد.

ایده‌ای که مدت زمان زیادی از ارائه‌ی آن نمی‌گذرد، فشرده‌سازی داده‌های ویدیویی مبتنی بر نواحی مطلوب (ROI) است. این ایده در کاربردهایی مورد استفاده قرار می‌گیرد که در آنها، تمام نواحی یک تصویر یا فریم از اهمیت یکسانی برخوردار نمی‌باشند. در این گونه تصاویر، نواحی بااهمیت‌تر با میزان فشرده‌سازی کمتر (و در نتیجه با میزان کیفیت بیشتر) و نواحی کم‌اهمیت‌تر با میزان فشرده‌سازی بیشتر (و در نتیجه با میزان کیفیت کمتر) کدگذاری می‌گردند. ایده‌ی کدگذاری مبتنی بر نواحی مطلوب در تصاویری مانند تصاویر پزشکی، سیگنالهای حیاتی، تصاویر ورزشی، و تصاویر پرسنلی قابل استفاده است زیرا در این گونه تصاویر تمام نواحی تصویر از یک درجه‌ی اهمیت برخوردار نبوده و برخی نواحی بیشتر از بقیه مورد توجه می‌باشند.

در این پایان نامه، روشی جهت فشرده‌سازی دنباله‌های ویدیویی ورزش فوتبال مبتنی بر نواحی مطلوب پیشنهاد شده است. در این راستا، سعی در شناسایی نواحی مطلوب و سپس کدگذاری مناسب هر ناحیه با توجه به میزان اهمیت آن داریم. در روش پیشنهادی از تبدیل موجک و کدگذاری ضرایب آن به کمک کدگذارهای SPC (به ویژه کدگذارهای SPIHT و EBCOT) استفاده شده است. مهمترین اجزا و یا تکنیکهای استفاده شده عبارتند از دسته‌بندی فریمها به دو نوع مستقل و وابسته و کدگذاری جداگانه‌ی هر کدام از آنها، استفاده از روش پیشنهادی جهت تعیین نوع هر فریم ورودی، استفاده از انطباق بلوکی در دو حوزه‌ی مکان و تبدیل، و نیز

استفاده از بازچینی بلوکها جهت افزایش کارایی کدگذارهای SPC، و بالاخره، تعریف و تعیین نواحی مطلوب در تصاویر ورزش فوتبال. کارایی فشرده‌سازی روش پیشنهادی با کارایی استاندارد MPEG4 و روش غیراستاندارد اما امروزی FLV برطبق معیارهای PSNR، بیت بر پیکسل (bpp)، و بیت بر ثانیه (bps) مقایسه شده است. نتایج انجام شبیه‌سازیها از برتری گاهاً قابل توجه روش پیشنهادی نسبت به روشهای مذکور حکایت دارد.

کلمات کلیدی: فشرده سازی ویدیو- نواحی مطلوب - ورزش فوتبال - نرخ بیت پایین

مقالات پذیرفته شده/چاپ شده‌ی مستخرج از پایان نامه

(۱) هادی گرایلو، مصطفی صفایی، رویا سلطانی، و آیدین خدائشناس "آشکارسازی برخط تغییر نما و تعیین نوع آن در دنباله های ویدئویی و پیاده سازی سخت افزاری آن در بستر پردازشگر سیگنال TMS320C5505"، چهارمین کنفرانس فناوری اطلاعات و دانش، دانشگاه صنعتی بابل، اردیبهشت ۱۳۹۱.

(۲) هادی گرایلو، مصطفی صفایی، و رویا سلطانی، "ارائه روشی برای آشکارسازی و تعیین نوع تغییر نما در دنباله های ویدئویی و مناسب برای پیاده سازی سخت افزاری زمان حقیقی در بستر FPGA و پردازنده های DSP" هفتمین کنفرانس ماشین بینایی و پردازش تصویر ایران، دانشگاه علم و صنعت ایران، آبان ۱۳۹۰.

(۳) محسن اکرمی، هادی گرایلو، و رویا سلطانی، " روشی کارا برای فشرده سازی تصاویر چهره پرسنلی در نرخ بیت های پایین به منظور استفاده در تشخیص هویت بلادرنگ" پذیرفته شده در اولین کنفرانس بازشناسی الگو و تحلیل تصویر ایران، زمان برگزاری: ۱۶ تا ۱۸ اسفند ۱۳۹۱، دانشگاه بیرجند.

[4] HadiGrailu, RoyaSoltani, and Mohsen Akrami, "ROI Soccer Video Compression Using Wavelet-Domain Block Matching and EBCOT Coding Techniques In Very Low Bit Rate Communication," Accepted in: The First Iranian conference on Pattern Recognition and Image Analysis, 6-8 March 2013, Birjand University.

[5] HadiGrailu, RoyaSoltani, and Mohsen Akrami, "Improving the Compression Efficiency of SPC Coders Based on Spectral Reordering and Block Matching Techniques," Accepted in: The 2012 5th International Conferences on Computer and Electrical Engineering (ICEEE 2012).

فهرست مطالب

فصل اول: مقدمه

- ۱-۱ مقدمه ----- ۱
- ۱-۱-۱ فشردگی‌سازی ویدیو و علت نیاز به آن ----- ۱
- ۲-۱-۱ فشردگی‌سازی مبتنی بر نواحی مطلوب ----- ۲
- ۳-۱-۱ فشردگی‌سازی دنباله‌های ویدیویی ورزشی ----- ۲
- ۲-۱ هدف و ساختار پایان نامه ----- ۳

فصل دوم: مروری بر استانداردهای موجود جهت فشردگی‌سازی ویدیو

- مقدمه ----- ۵
- ۱-۲ کاهش تزیاید مکانی ----- ۶
- ۱-۱-۲ کدگذاری پیش‌بین ----- ۶
- ۲-۱-۲ کدگذاری مبتنی بر تبدیل ----- ۸
- ۲-۲ کاهش تزیاید زمانی ----- ۱۰
- ۳-۲ کدک ویدیویی عمومی بین فریمی ----- ۱۵
- ۱-۳-۲ حلقه بین فریمی ----- ۱۵
- ۲-۳-۲ تخمین گر حرکت ----- ۱۵
- ۳-۳-۲ سوئیچ بین/درون ----- ۱۶
- ۴-۳-۲ تبدیل DCT ----- ۱۶
- ۵-۳-۲ بافر ----- ۱۶

- ۱۷ ----- ۶-۳-۲ واحد رمزگشا
- ۱۷ ----- ۴-۲ کدگذاری به منظور کنفرانس ویدیویی (H.261)
- ۱۸ ----- ۱-۴-۲ قالب و ساختار ویدیو
- ۲۰ ----- ۲-۴-۲ الگوریتم کدگذاری ویدیو
- ۲۱ ----- ۱-۲-۴-۲ پیش بینی
- ۲۲ ----- ۲-۲-۴-۲ تصمیم در مورد انجام جبران حرکت
- ۲۳ ----- ۳-۲-۴-۲ تصمیم در مورد درون/بین فریمی
- ۲۴ ----- ۴-۲-۴-۲ کدگذاری با طول متغیر دوبعدی
- ۲۵ ----- ۵-۲-۴-۲ حلقه فیلتر
- ۲۷ ----- ۳-۴-۲ کنترل نرخ
- ۲۷ ----- ۵-۲ کدگذاری به منظور ذخیره اطلاعات ویدیویی (MPEG-1)
- ۲۸ ----- ۱-۵-۲ دوباره چینی تصویر
- ۳۱ ----- ۲-۵-۲ ساختار ویدیو
- ۳۱ ----- ۱-۲-۵-۲ گروه تصاویر
- ۳۲ ----- ۲-۲-۵-۲ تصویر
- ۳۲ ----- ۳-۲-۵-۲ قطعه
- ۳۳ ----- ۴-۲-۵-۲ ماکروبلوک
- ۳۳ ----- ۵-۲-۵-۲ بلوک
- ۳۴ ----- ۳-۵-۲ واحد رمزگذار
- ۳۶ ----- ۴-۵-۲ ماتریس وزن دهی کوانتیزاسیون
- ۳۶ ----- ۵-۵-۲ واحد رمزگشا

- ۳۷ ----- کدگذاری اطلاعات ویدیویی دیجیتال با کیفیت بالا (MPEG-2)
- ۳۸ ----- ۱-۶-۲ پروفایلها و سطوح
- ۴۰ ----- ۲-۶-۲ تفاوتهای MPEG-1 با MPEG-2
- ۴۰ ----- ۱-۲-۶-۲ تفاوتهای اصلی
- ۴۰ ----- ۲-۲-۶-۲ تفاوتهای فرعی
- ۴۱ ----- ۷-۲ کدگذاری اطلاعات ویدیویی در کاربرد مخابرات نرخ پایین (H.263)
- ۴۲ ----- ۱-۷-۲ تفاوتهای H.263 با H.261 و MPEG-1
- ۴۴ ----- ۸-۲ کدگذاری ویدیو مبتنی بر محتوا (MPEG-4)
- ۴۵ ----- ۱-۸-۲ پروفایلها و سطوح
- ۴۵ ----- ۲-۸-۲ صفحه اشیاء ویدیویی (VOP)
- ۴۶ ----- ۳-۸-۲ کدگذاری اشیاء
- ۴۶ ----- ۱-۳-۸-۲ کدگذاری VOP ها
- ۴۸ ----- ۲-۳-۸-۲ تشکیل VOP ها
- ۴۸ ----- ۳-۳-۸-۲ ناحیه بندی تصویر
- ۴۸ ----- ۴-۳-۸-۲ کدگذاری شکل
- ۴۹ ----- ۹-۲ کدگذاری ویدیویی پیشرفته (H.264)
- ۵۰ ----- ۱-۹-۲ قالب تصویر
- ۵۰ ----- ۲-۹-۲ قطعه سازی
- ۵۱ ----- ۱۰-۲ توصیف محتوا، جستجو و حمل (MPEG-7 و MPEG-21)

فصل سوم: مبانی تبدیل موجک

- ۵۳ ----- ۱-۳ چرا تبدیل موجک؟
- ۵۴ ----- ۲-۳ کدگذاری زیرباندی
- ۵۷ ----- ۳-۳ تبدیل موجک
- ۵۹ ----- ۱-۳-۳ تبدیل موجک گسسته
- ۶۰ ----- ۲-۳-۳ نمایش چند درجه تفکیک
- ۶۵ ----- ۳-۳-۳ تبدیل موجک و فیلتر بانک
- ۶۶ ----- ۴-۳-۳ سیستمهای مرتبه بالاتر
- ۶۸ ----- ۴-۳ کدگذاری زیرتصویرهای موجک
- ۶۹ ----- ۱-۴-۳ کوانتیزه کردن با تقریب متوالی
- ۷۱ ----- ۲-۴-۳ شباهتهای بین باندها

فصل چهارم: مبانی کدگذارهای EZW، SPIHT، و EBCOT

- ۷۳ ----- ۱-۴ الگوریتم EZW
- ۷۶ ----- ۱-۱-۴ تحلیل الگوریتم EZW
- ۷۸ ----- ۲-۴ روش بخش‌بندی مجموعه به صورت درختهای سلسله مراتبی (SPIHT)
- ۸۱ ----- ۱-۲-۴ الگوریتم کدگذاری
- ۸۵ ----- ۳-۴ کدگذاری بلوکی جاسازی شده با برش بهینه شده (EBCOT)
- ۸۸ ----- ۱-۳-۴ کدگذاری صفحه بیتی
- ۸۹ ----- ۲-۳-۴ کدگذاری صفحه‌های بیتی با استفاده از کدگذاری حسابی شرطی
- ۹۱ ----- ۳-۳-۴ کدگذاری صفحه بیتی کسری

- ۹۳ ----- ۱-۳-۳-۴ مرحله انتشار اهمیت
- ۹۶ ----- ۲-۳-۳-۴ مرحله اصلاح اندازه
- ۹۷ ----- ۳-۳-۳-۴ مرحله پاکسازی
- ۹۸ ----- ۴-۳-۴ تشکیل لایه و سازمان‌دهی دنباله‌ی بی‌تی
- ۹۹ ----- ۵-۳-۴ کنترل نرخ
- ۱۰۰ ----- ۴-۴ مثالی از نحوه‌ی کدگذاری
- ۱۰۰ ----- ۱-۴-۴ مقدمه
- ۱۰۱ ----- ۲-۴-۴ مثال استفاده شده
- ۱۰۳ ----- ۳-۴-۴ کدگذاری اولین صفحه بی‌تی
- ۱۰۳ ----- ۱-۳-۴-۴ مرحله پاکسازی
- ۱۰۶ ----- ۴-۴-۴ کدگذاری دومین صفحه بی‌تی
- ۱۰۷ ----- ۱-۴-۴-۴ مرحله انتشار اهمیت
- ۱۰۹ ----- ۲-۴-۴-۴ مرحله اصلاح اندازه
- ۱۱۰ ----- ۳-۴-۴-۴ مرحله پاکسازی
- ۱۱۰ ----- ۵-۴-۴ ادامه و پایان کدگذاری

فصل پنجم: روش پیشنهادی

- ۱۱۲ ----- ۱-۵ مقدمه
- ۱۱۲ ----- ۲-۵ پایگاه داده‌ی مورد استفاده
- ۱۱۳ ----- ۳-۵ روش پیشنهادی
- ۱۱۶ ----- ۱-۳-۵ تعیین نوع فریم‌ها

- ۱۱۶ ----- ۲-۳-۵ پیش پردازش پیشنهادی
- ۱۱۹ ----- ۳-۳-۵ کاهش تزیاید به کمک انطباق بلوکی
- ۱۲۴ ----- ۴-۳-۵ تعیین نواحی مطلوب
- ۱۲۶ ----- ۵-۳-۵ دیگر حالت‌های کاری
- ۱۲۷ ----- ۴-۵ نتیجه‌گیری

فصل ششم: نتایج شبیه‌سازیها

- ۱۲۸ ----- ۱-۶ مقدمه
- ۱۲۸ ----- ۲-۶ بررسی میزان تاثیر تکنیک‌های انطباق بلوکی و بازچینی بلوکها روی کارایی کدگذارهای SPC
- ۱۳۳ ----- ۳-۶ بررسی کارایی فشرده‌سازی روش پیشنهادی
- ۱۳۵ ----- ۴-۶ بررسی میزان تاثیر حالت‌های کاری اختیاری
- ۱۳۸ ----- ۵-۶ نتیجه‌گیری

فصل هفتم: نتیجه‌گیری و پیشنهاد راهکارهای آینده

- ۱۳۹ ----- ۱-۷ نتیجه‌گیری
- ۱۴۱ ----- ۲-۷ پیشنهاد راهکارهای آینده
- ۱۴۱ ----- مراجع

فهرست جداول

فصل دوم: مروری بر استانداردهای موجود جهت فشرده‌سازی ویدیو

- ۱-۲ پروفایل‌های مختلف تعریف شده در MPEG-2 ----- ۲۸
- ۲-۲ سطوح تعریف شده برای هر پروفایل در MPEG-2 ----- ۳۹
- ۴-۲ پنج قالب تصویری مورد پشتیبانی H.263 ----- ۴۲

فصل چهارم: مبانی کدگذارهای EZW، SPIHT، و EBCOT

- ۱-۴ نحوه انتخاب هر یک از ۹ فحوا مبتنی بر اهمیت همسایگان ----- ۹۴
- ۲-۴ علامت پیش‌بینی شده، \hat{x} ، و فحواهای κ^{SC} واحد پایه SC ----- ۱۰۴
- ۳-۴ فحوا κ^{ZC} از واحد پایه ZC . نماد X به معنای هرمقداری است ----- ۱۰۵
- ۴-۴ کدگذاری صفحه بیتی $n=5$ به کمک مرحله‌ی پاکسازی ----- ۱۰۷
- ۵-۴ کدگذاری صفحه بیتی $n=4$ توسط مرحله‌ی انتشار اهمیت ----- ۱۰۹
- ۶-۴ فحواهای κ^{MR} مورد استفاده در واحد پایه MR ----- ۱۱۰
- ۷-۴ کدگذاری صفحه بیتی $n=4$ توسط مرحله‌ی پاکسازی ----- ۱۱۱

فصل ششم: نتایج شبیه‌سازی

- ۱-۶ میانگین PSNR (برحسب dB) و واریانس آن (برحسب درصد نسبت به میانگین) در نرخ بیت ۰/۱ bpp برای
- تصاویر استفاده شده و دسته‌ی اول آزمایشها (بدون اعمال هرکدام از دو بهبود پیشنهادی) ----- ۱۳۱
- ۲-۶ میانگین PSNR (برحسب dB) و واریانس آن (برحسب درصد نسبت به میانگین) در نرخ بیت ۰-۱ bpp ----- ۱۳۱
- ۳-۶ میانگین PSNR (برحسب dB) و واریانس آن (برحسب درصد نسبت به میانگین) در نرخ بیت ۰/۱ bpp برای

تصاویر استفاده شده و دسته‌ی سوم آزمایشها (اعمال تکنیک بازچینی بلوکها)----- ۱۳۲

۴-۶ میانگین PSNR (برحسب dB) و واریانس آن (برحسب درصد نسبت به میانگین) در نرخ بیت ۰/۱ bpp برای

تصاویر استفاده شده و دسته‌ی چهارم آزمایشها (اعمال هر دو بهبود پیشنهادی)----- ۱۳۲

۵-۶ مقدار متوسط PSNR (برحسب dB) و واریانس (برحسب درصدی از مقدار متوسط) روش پیشنهادی و دو روش

۱۳۵ ----- ۳۲ Kbps ثابت FLV و MPEG4 در نرخ بیت ارسال

۶-۶ مقدار کاهش درصدی نرخ بیت در نتیجه‌ی استفاده از یکی یا تمام حالت‌های کاری پیشنهادی نسبت به حالت

اصلی (یعنی عدم استفاده از هر گونه حالت کاری)----- ۱۳۷

فهرست شکل ها

فصل دوم: مروری بر استانداردهای موجود جهت فشرده‌سازی ویدیو

- ۱-۲ دیاگرام بلوکی یک کدک DPCM----- ۷
- ۲-۲ رخداد همزمان یک جفت از پیکسل‌ها----- ۸
- ۳-۲ فریمهای فعلی و قبلی مورد استفاده در یک پنجره جستجو----- ۱۱
- ۴-۲ مثالی از روش جستجوی CSA برای حالت (پیکسل بر فریم) $w=8$ ----- ۱۳
- ۵-۲ یک هرم تصویر سه سطحی----- ۱۴
- ۶-۲ یک کد گذار پیش بین بین فریمی عام----- ۱۵
- ۷-۲ دیاگرام بلوکی یک رمز گشا----- ۱۷
- ۸-۲ دیاگرام بلوکی یک رمز گذار صوتی-تصویری H.261----- ۱۸
- ۹-۲ مفاهیم بلوک، ماکروبلوک و GOB در تصاویری با قالب CIF و QCIF----- ۲۰
- ۱۰-۲ دیاگرام بلوکی کدگذار ویدیویی H.261----- ۲۰
- ۱۱-۲ تصمیم گیری در مورد استفاده/عدم استفاده از جبران حرکت----- ۲۲
- ۱۲-۲ تصمیم گیری در مورد انتخاب حالت‌های بین/درون فریمی----- ۲۳
- ۱۳-۲ نحوه جاروب ضرایب تبدیل 8×8 ----- ۲۳
- ۱۴-۲ تصویری با نام Calire (الف) اصلی، (ب) کدگذاری شده با H.261 و در نرخ 256 kbit/sec ----- ۲۶
- ۱۵-۲ تصویر Claire کدگذاری شده توسط H.261 و با نرخ (الف) 128 kb/s ، (ب) 64 kb/s ----- ۲۶
- ۱۶-۲ تصویر کدگذاری شده با استفاده از فیلتر حلقه در حالت‌های (الف) 128 kb/s ، (ب) 64 kb/s ----- ۳۱

- ۱۷-۲ مثالی از یک GOP در MPEG-1 ----- ۳۱
- ۱۸-۲ ساختار کدگذاری ویدیو در MPEG-1 ----- ۳۴
- ۱۹-۲ شمای ساده شده‌ی رمزگذار در MPEG-1 ----- ۳۵
- ۲۰-۲ ماتریس کوانتیزاسیون ----- ۳۶
- ۲۱-۲ دیاگرام بلوکی رمزگشای MPEG-1 ----- ۴۱
- ۲۲-۲ نوع جاروب (الف) زیگزاگ، (ب) متناوب ----- ۴۳
- ۲۳-۲ پیش بینی بردار جابجایی ----- ۴۴
- ۲۴-۲ مثالی از صفحه اشیاء ویدیویی (VOP) ----- ۴۶
- ۲۵-۲ استفاده از ماسک جهت نمایش اشیاء ----- ۴۶
- ۲۶-۲ دیاگرام بلوکی کدک مورد استفاده در مدل تایید، (الف) رمزگذار، (ب) رمزگشا ----- ۴۷
- ۲۷-۲ ساختار کدگذار ویدیویی ----- ۴۸
- ۲۸-۲ مثالی از نحوه‌ی تشکیل یک VOP ----- ۴۸
- ۲۹-۲ نمونه‌ای از یک قطعه ----- ۵۱

فصل سوم: مبانی تبدیل موجک

- ۱-۳ یک نمونه بانک فیلترهای میان گذر ----- ۵۷
- ۲-۳ تاثیر اتساع و جابجایی زمانی روی موجک مادر؛ (الف) موجک مادر $\Psi_{1,0}(t) = \Psi(t)$ با $a=1$ و $b=0$ ؛ (ب) حالت $a=1$ و $b \neq 0$ ؛ (ج) حالت $a=2$ و $b=0$ ؛ و (د) حالت $a=1/2$ و $b=0$ ----- ۵۹
- ۳-۳ فضاهای چند درجه‌ی تفکیک ----- ۶۳
- ۴-۳ (الف) تابع مقیاس هار، (ب) موجک هار، (ج) تقریب یک تابع پیوسته، $X(t)$ ، در مقیاس درشت‌تر $A\phi(x(t))$ و (د) مقیاس نرمتر ----- ۶۴
- ۵-۳ یک مرحله تبدیل موجک، (الف) تحلیل (تجزیه)، (ب) ترکیب ----- ۶۷

- ۶-۳ تبدیل موجک چندباندی به کمک تکرار تجزیه‌ی دوباندی ----- ۶۷
- ۷-۳ (الف) هفت زیرتصویر تولید شده توسط فرآیند نشان داده شده در شکل ۶-۳، (ب) چیدمان باندهای مختلف متناظر با فرایند ۶-۳ ----- ۶۸
- ۸-۳ ایده‌ی اساسی روش تقریب متوالی ----- ۷۰
- ۹-۳ نمایش درخت تربیعی از باندهای متعلق به یک راستا ----- ۷۲

فصل چهارم: مبانی کدگذارهای EZW، SPIHT، و EBCOT

- ۱-۴ نمایشی از یک SOT و نحوه‌ی بخش بندی در روش SPIHT ----- ۸۰
- ۲-۴ کوانتایزر یکنواخت منطقه مرده با اندازه گام Δ_b ----- ۸۸
- ۳-۴ همسایه مورد استفاده در فحوای مورد نیاز کدگذار آنتروپی ----- ۹۰
- ۴-۴ ترتیب جاروب نواری در یک بلوک کد مفروض ----- ۹۱
- ۵-۴ تاثیر ترتیب کدگذاری صفحه بیتی کسری روی میزان کاهش اعوجاج ----- ۹۲
- ۶-۴ منحنی نرخ بیت-اعوجاج در حالت بهینه سازی ----- ۹۳
- ۷-۴ بازدهی فشرده‌سازی الگوریتمهای مختلف کدگذاری ضرایب موجک ----- ۱۰۰
- ۸-۴ بخشی از بلوک کد مورد استفاده در مثال مورد بررسی ----- ۱۰۲
- ۹-۴ صفحه‌ی بیتی $n=5$ ----- ۱۰۳
- ۱۰-۴ صفحه بیتی $n=4$ ----- ۱۰۷
- ۱۱-۴ مرحله‌ی انتشار اهمیت روی صفحه بیتی $n=4$ ----- ۱۰۸
- ۱۲-۴ مرحله پاکسازی روی صفحه بیتی $n=4$ ----- ۱۱۰

فصل پنجم: روش پیشنهادی

- ۱-۵ کارت دریافت سیگنالهای تلویزیونی مدل Miracle نوع هیبرید (آنالوگ/دیجیتال) و نحوه‌ی اتصال آن به رایانه از طریق درگاه USB-----۱۱۳
- ۲-۵ روندنمای روش پیشنهادی جهت فشرده‌سازی دنباله‌های ویدیویی ورزشی -----۱۱۳
- ۳-۵ مرتب کردن بلوکها به ترتیب نزولی، تغییر مکان بلوکهای اولیه و تشکیل تصویر جدید -----۱۱۷
- ۴-۵ (الف) تصویر اصلی، (ب) نتیجه‌ی فرآیند پیش پردازشی پیشنهادی-----۱۱۸
- ۵-۵ انطباق بلوکی در حالت دو تصویر ورودی، (الف) فریم فعلی، (ب) فریم قبلی -----۱۱۹
- ۶-۵ انطباق بلوکی در حالت یک تصویر ورودی؛ در این حالت تصویر ورودی ماتریس ضرایب تبدیل موجک-----۱۲۰
- ۷-۵ یک نمونه ماتریس تبدیل موجک (الف) و نتیجه انطباق بلوکی (ب) روی ماتریس (الف)-----۱۲۲
- ۸-۵ الگوی بلوکی روش جستجوی سه مرحله‌ای-----۱۲۳
- ۹-۵ الگوی بلوکی روش جستجوی پیشنهادی تک مرحله‌ای-----۱۲۳
- ۱۰-۵ (الف و ب) دو فریم متوالی، (ج) تصویر حاصل از تفریق دو فریم (الف) و (ب)-----۱۲۵
- ۱۱-۵ نمونه نتیجه‌ی خوشه‌یابی با سه خوشه و انتخاب بزرگترین خوشه به عنوان ناحیه‌ی زمین بازی فوتبال-----۱۲۵

فصل اول

مقدمه

۱-۱ مقدمه

۱-۱-۱ فشردن سازی ویدیو و علت نیاز به آن

امروزه در سازمانهای دولتی/غیر دولتی بسیاری در اقصی نقاط دنیا، داده‌های ویدیویی بسیاری به صورت روزافزون در حال تولید، ذخیره، و ارسال است. نگهداری یا ذخیره‌ی چنین حجم انبوهی از دنباله‌های ویدیویی مستلزم استفاده از حافظه‌های با حجم بسیار زیاد است که چندان مقرون به صرفه نبوده و علاوه بر این حجم فیزیکی نسبتاً زیادی اشغال می‌شود که منجر به مشکل شدن جابجایی و دسترسی به داده‌های مذکور می‌شود. یک راه برای غلبه نسبی بر مشکل نگهداری یا ذخیره‌ی دنباله‌های ویدیویی استفاده از روشهای موثر فشردن سازی ویدیو می‌باشد. به این ترتیب حجم داده‌های ویدیویی کاهش قابل ملاحظه‌ای یافته و مدیریت آنها موثرتر می‌گردد. به علاوه از آنجا که روشهای فشردن سازی در دو نوع بااتلاف و بدون اتلاف وجود دارند می‌توان از روش مناسب جهت رسیدن به میزان فشردن سازی و میزان کیفیت مطلوب استفاده کرد. برخی کاربردها نیاز چندان به کیفیت بالا نداشته و بنابراین، می‌توانند میزان فشردن سازی را افزایش دهند.

۱-۱-۲ فشردگی‌سازی مبتنی بر نواحی مطلوب

ایده‌ای که مدت زمان زیادی از ارائه‌ی آن نمی‌گذرد، فشردگی‌سازی داده‌های ویدیویی مبتنی بر نواحی مطلوب (ROI)^۱ است. این ایده در کاربردهایی مورد استفاده قرار می‌گیرد که در آنها، تمام نواحی یک تصویر یا فریم از اهمیت یکسانی برخوردار نمی‌باشند. در این گونه تصاویر، نواحی بااهمیت‌تر با میزان فشردگی‌سازی کمتر (و در نتیجه با میزان کیفیت بیشتر) و نواحی کم‌اهمیت‌تر با میزان فشردگی‌سازی بیشتر (و در نتیجه با میزان کیفیت کمتر) کدگذاری می‌گردند. ایده‌ی کدگذاری مبتنی بر نواحی مطلوب در تصاویری مانند تصاویر پزشکی، سیگنال‌های حیاتی، تصاویر ورزشی، و تصاویر پرسنلی قابل استفاده است زیرا در این گونه تصاویر تمام نواحی تصویر از یک درجه‌ی اهمیت برخوردار نبوده و برخی نواحی بیشتر از بقیه مورد توجه می‌باشند.

۱-۱-۳ فشردگی‌سازی دنباله‌های ویدیویی ورزشی

امروزه، مسابقات ورزشی بسیاری در اقصی نقاط دنیا برگزار می‌گردد. رسانه‌های مختلف خبری به منظور پوشش خبری این ورزشها تمام مدت مسابقات را ذخیره و سپس ارسال می‌کنند. بنابراین، دنباله‌های ویدیویی ورزشی با حجم انبوه و به صورت روزافزون در حال تولید، ذخیره، و ارسال است. با توجه به دلایلی که در زیربخشهای قبلی آمد، نیاز بسیاری به فشردگی‌سازی این داده‌های ویدیویی می‌باشد. هرچه روش فشردگی‌سازی موثرتر باشد، کارایی و سودمندی آن نیز بیشتر خواهد بود.

در بین ورزشهای موجود، به نظر می‌رسد ورزش فوتبال از اهمیت، توجه و طرفداری خاصی برخوردار باشد. از آنجا که طبیعت و ماهیت تصاویر/فریمهای ورزش فوتبال به گونه‌ای است که تمام نواحی یک صحنه‌ی ورزشی مورد توجه نمی‌باشند بلکه بیشتر نواحی‌ای که توپ یا بازیکنان فعال در آنها حضور داشته باشند مورد توجه هستند، بنابراین، استفاده از روشهای فشردگی‌سازی مبتنی بر نواحی مطلوب در مورد دنباله‌های ویدیویی ورزش

¹Region Of Interest (ROI)

فوتبال منطقی به نظر می‌رسد. طبق دانسته‌ی ما تاکنون، توجه خاصی به فشردگی‌سازی این نوع از دنباله‌های ویدیویی نگردیده است.

۱-۲ هدف و ساختار پایان نامه

در این پایان نامه، به دنبال ارائه روشی جهت فشردگی‌سازی دنباله‌های ویدیویی ورزش فوتبال مبتنی بر نواحی مطلوب می‌باشیم. در این راستا سعی در شناسایی نواحی مطلوب و سپس کدگذاری مناسب هر ناحیه با توجه به میزان اهمیت آن داریم. در فشردگی‌سازی نواحی مختلف، عموماً از تبدیل موجک و کدگذاری ضرایب آن استفاده می‌کنیم. به دلیل ویژگیهای خاصی که زیرباندهای مختلف حاصل از تبدیل موجک دارند، ضرایب این تبدیل به شیوه‌ای متفاوت و موثرتر از شیوه‌های عمومی کدگذاری مانند کدگذاری هافمن^۲ و حسابی^۳، کدگذاری می‌گردند. از جمله‌ی این روشهای جدید می‌توان به روشهای EZW، SPIHT، و EBCOT اشاره کرد.

ساختار این پایان نامه به این صورت است که در فصل دوم، استانداردهای فشردگی‌سازی موجود مورد بررسی قرار گرفته‌اند. از آنجا که طبق دانسته‌ی ما روشی جهت فشردگی‌سازی دنباله‌های ویدیویی ورزش فوتبال موجود نیست، روشهای فشردگی‌سازی استاندارد مورد بررسی قرار گرفته‌اند زیرا از درجه‌ی اعتبار بیشتری نسبت به روشهای غیراستاندارد برخوردار هستند گرچه این استانداردها از نظر به روز بودن و جدیدتر بودن، عقب‌تر از روشهای غیر استاندارد و جدید امروزی هستند. علت دیگری که ما در فصل دوم، استانداردهای موجود را بررسی کرده‌ایم این است که معیاری جهت بررسی عملکرد فشردگی‌سازی روش پیشنهادی ما باشند. در فصل سوم مفاهیم پایه‌ای تبدیل موجک و در فصل چهارم نیز مفاهیم و اساس مهمترین روشهای کدگذاری موجود امروزی جهت کدگذاری ضرایب موجک‌بررسی و تشریح خواهند شد. تبدیل موجک و کدگذاری ضرایب آن نقش مهمی در روش پیشنهادی ما دارند لذا برای هر یک از آنها فصلی جداگانه در نظر گرفته شده است. در فصل پنجم،

²Huffman Coding

³Arithmetic Coding

روش پیشنهادی خود را معرفی کرده و اساس آن را توضیح می‌دهیم. عملکرد و کارایی فشرده‌سازی روش پیشنهادی در فصل ششم مورد بررسی و مقایسه با چند استاندارد موجود قرار خواهد گرفت. فصل هفتم به نتیجه‌گیری و ارائه چند پیشنهاد جهت ادامه کار می‌پردازد.

فصل دوم

مروری بر استانداردهای موجود جهت فشرده سازی

ویدیو

مقدمه

تحلیل آماری سیگنالهای ویدیویی نشان می‌دهد که همبستگی و شباهت زیادی بین فریمهای متوالی و نیز بین عناصر داخلی هر فریم وجود دارد. حذف این همبستگی‌ها منجر به فشرده سازی در پهنای باند می‌شود بدون اینکه تاثیر چندان نامطلوبی در درجه تفکیک^۴ تصاویر برجای بگذارد. به علاوه، سیستم بینایی انسان (HVS^۵) به حذف برخی اطلاعات مکانی-زمانی^۶ سیگنال ویدیویی غیرحساس است و از این رو می‌توان از آن برای افزایش میزان فشرده سازی سیگنال ویدیویی استفاده کرد.

هرگاه فقط همبستگی‌های داخلی (یعنی همبستگی‌های مکانی) یک تصویر حذف شود، به آن اصطلاح کدگذاری درون-فریمی^۷ اطلاق می‌شود مانند استاندارد JPEG، اما اگر همبستگی‌های بین-فریمی (یعنی همبستگی‌های زمانی) نیز حذف شود، به آن کدگذاری بین-فریمی^۸ اطلاق می‌گردد. نوع متداولی از این نوع کدگذاری،

^۴Resolution

^۵Human Visual System

^۶Spatio-Temporal

^۷Intra-Frame Coding

^۸Inter-Frame Coding

کدگذاری بین-فریمی پیش‌بین^۹ می‌باشد که در استانداردهایی نظیر H.261، H.263، H.264، MPEG1، MPEG2 و MPEG4 مورد استفاده قرار می‌گیرد. این نوع از کدگذاری مبتنی بر سه اصل زیر به منظور کاهش تزیاید^{۱۰} می‌باشد:

- ۱- کاهش تزیاید مکانی: به منظور کاهش تزیاید و یا شباهت بین پیکسل‌های هر فریم
 - ۲- کاهش تزیاید زمانی: به منظور کاهش شباهت بین فریم‌های متوالی از طریق کدگذاری تفاضل آنها
 - ۳- کدگذاری آنتروپی: به منظور کاهش تزیاید بین نمادهای حاصل از مراحل فشرده‌سازی فوق با استفاده از تکنیک‌های کدگذاری با طول متغیر (VLC^{۱۱})
- در ادامه هر یک از این سه اصل توضیح داده می‌شود.

۱-۲ کاهش تزیاید مکانی

۱-۱-۲ کدگذاری پیش‌بین

در این روش معمولاً مقدار هر پیکسل فعلی بر حسب مقادیر پیکسل‌های از قبل کدگذاری شده توصیف و یا پیش‌بینی می‌شود. سپس خطای این پیش‌بینی کدگذاری می‌شود. چنین روشی به نام DPCM^{۱۲} شناخته می‌شود.

شکل ۱-۲ بلوک دیاگرام مربوط به کدگذار و کدگشای DPCM را نشان می‌دهد. به ترکیب کدگذار و کدگشا، کدک (CODEC^{۱۳}) گفته می‌شود. در واحد کدگشا، سیگنال خطای دریافتی با مقادیر پیش‌بینی شده جمع شده تا سیگنال اصلی بازسازی شود.

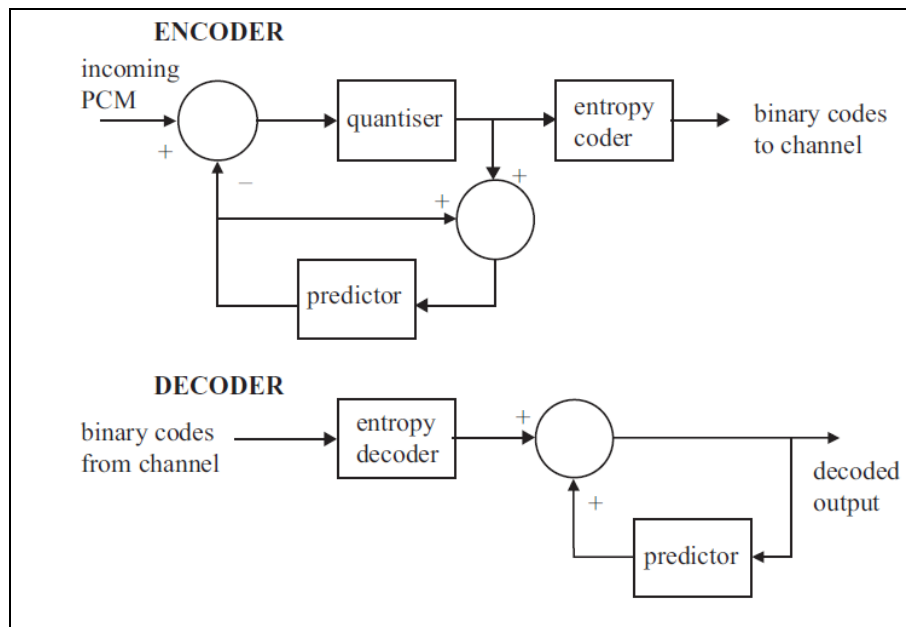
⁹Inter-Frame Predictive Coding

¹⁰Redundancy

¹¹Variable Length Coding

¹²Differential Pulse Code Modulation

¹³Coder/Decoder



شکل ۱-۲ دیاگرام بلوکی یک کدک DPCM

در طرح فوق، واحد چندی-ساز (یا کوانتایزر^{۱۴}) عامل اصلی فشرده‌سازی با اتلاف^{۱۵} است. اگر از این واحد استفاده نشود، نوع فشرده‌سازی، بدون اتلاف^{۱۶} خواهد بود.

برای انجام پیش‌بینی مقدار پیکسل فعلی می‌توان از پیکسل‌های متعلق به فریم فعلی و یا فریم قبلی و یا ترکیب این دو استفاده کرد که به ترتیب به نام‌های کدگذار پیش‌بین درون-فریمی، کدگذار پیش‌بین بین-فریمی، و کدگذار پیش‌بین ترکیبی شناخته می‌شوند. از آنجا که حداقل تعداد بیت‌هایی که برای کدگذاری هر مقدار خطای پیش‌بینی، ۱ بیت است، بنابراین، روش DPCM برای کاربردهای نرخ بیت پایین چندان مناسب نمی‌باشد. البته اگر به جای هر یک پیکسل، گروهی از پیکسل‌ها را کدگذاری کنیم، می‌توان به نرخ بیت‌های پایین‌تر از ۱ بیت دست یافت. یک نمونه از چنین روش‌های مبتنی بر کدگذاری گروهی پیکسل‌ها، کدگذاری بلوکی مبتنی بر تبدیل است. علیرغم مزیت کدگذاری بلوکی، امروزه به دلایلی همچنان از DPCM در روش‌های فشرده‌سازی ویدیو استفاده می‌شود. برای مثال سرعت عملکرد DPCM بیشتر از کدگذاری بلوکی بین-فریمی است. همچنین در

¹⁴Quantizer

¹⁵ Lossy Compression

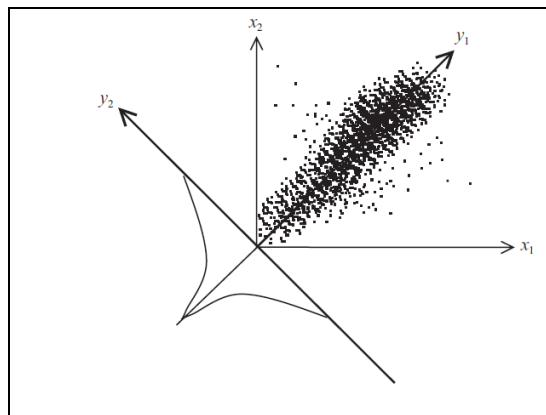
¹⁶Lossless Compression

کدگذاری بردارهای حرکت مربوط به یک شیء متحرک که در یک راستای مشخص در حال حرکت باشد، کدگذاری تفاضلات بردارهای جابجایی مربوطه تا حد خوبی حجم اطلاعات مربوطه را کاهش می‌دهد.

۲-۱-۲ کدگذاری مبتنی بر تبدیل

در این دسته از روشها که بیشتر به منظور کاهش تزايد مکانی پیکسلهای یک تصویر استفاده می‌شوند، قبل از کاهش حجم داده‌ها، ابتدا پیکسلها را به حوزه دیگری تبدیل می‌کنند. اهمیت استفاده از تبدیل از اینجا آشکار می‌شود که عمده‌ی انرژی بیشتر تصاویر طبیعی در بازه‌ی فرکانسهای پایین قرار دارد. در نتیجه عمده انرژی سیگنال تصویر تنها با چند ضریب محدود قابل بازسازی است. در مرحله‌ی بعد معمولاً این ضرایب تبدیل کوانتیزه می‌شوند تا ضرایب کم‌اهمیت که تاثیر ناچیزی روی کیفیت تصویر دارند، حذف شوند. البته استفاده از کوانتایزر، همان طور که قبلاً نیز بیان شد، منجر به فشرده‌سازی با اتلاف می‌گردد.

برای بررسی بهتر عملکرد کدگذاری مبتنی بر تبدیل شکل ۲-۲ را در نظر بگیرید که مربوط به رخداد (یا وقوع) همزمان دو پیکسل همسایه x_1 و x_2 می‌باشد.



شکل ۲-۲ رخداد همزمان یک جفت از پیکسلها

هر کدام از پیکسلهای x_1 و x_2 می‌توانند هر مقدار دلخواهی بین ۰ تا ۲۵۵ اخذ کنند اما از آنجا که شباهت و همبستگی بین این دو معمولاً زیاد است، وقوع همزمان آنها با احتمال زیادی روی یک خط با شیب ۴۵ درجه رخ خواهد داد. بنابراین اگر مختصات x_1x_2 را به اندازه ۴۵ درجه بچرخانیم تا به مختصات جدید y_1y_2 برسیم،

انتظار داریم مطابق با شکل ۲-۲ پراکندگی پیکسلها حول y_2 نسبتاً متقارن و نسبت به صفر متمرکز باشد. همچنین اندازه‌ی مولفه‌ها در راستای y_2 کوچک باشند (به دلیل وجود تزیاید و نزدیک بودن محل هندسی به خط ۴۵ درجه). تعداد بیت‌های لازم برای کدگذاری مولفه‌ی y_1 حدوداً همان تعداد لازم برای کدگذاری مولفه‌های x_1 و x_2 است اما مطمئناً تعداد بیت‌های لازم برای کدگذاری مولفه‌ی y_2 کمتر از تعداد لازم برای کدگذاری مولفه‌های x_1 و x_2 است (زیرا شامل اعداد کوچکتری است).

حال اگر به جای دو پیکسل، گروه‌های بزرگتری از پیکسلها، برای مثال گروه‌های N تایی، را در نظر بگیریم، در اثر انجام تبدیل، مولفه‌های y_1 روی قطر اصلی یک ابرکره قرار خواهند گرفت و بنابراین شامل اعداد بسیار کوچکی خواهند بود. دیگر مولفه‌های y_2 تا y_N فقط شامل اطلاعات باقیمانده هستند.

نتیجه این که استفاده از گروه‌های بزرگتری از پیکسلها منجر به میزان فشرده‌سازی بیشتری خواهد شد. البته این که بالاترین مقدار N که همچنان منجر به افزایش فشرده‌سازی شود چه قدر است بستگی به شعاع مکانی برقراری همبستگی بین پیکسلهای تصویر دارد.

یک حالت ایده آل برای ماتریس تبدیل ماتریسی است که به طور کامل ضرایب تبدیل را ناهمبسته کند. بنابراین اگر R_{xx} ماتریس کواریانس منبع (پیکسلهای) ورودی، x ، باشد عناصر ماتریس تبدیل T به گونه‌ای محاسبه می‌شوند که کواریانس ضرایب $R_{yy} = TR_{xx}T'$ یک ماتریس قطری گردد. تبدیلی که بر این مبنا ایجاد گردد همان تبدیل معروف KLT است [۱]. گرچه این تبدیل بهینه است و بیشترین مقدار فشرده‌سازی را به دست می‌دهد اما برای کاربرد فشرده‌سازی مناسب نیست زیرا با تغییر مشخصات آماری تصویر، ماتریس تبدیل باید مجدداً محاسبه گردد. بنابراین علاوه بر مشکل بالا بودن حجم محاسبات باید عناصر ماتریس تبدیل نیز برای واحد رمزگشا ارسال گردد.

یک گزینه بهتر برای ماتریس تبدیل استفاده از تبدیل DCT است. مشخصات نرخ فشرده‌سازی-عوجاج آن تا حد زیادی نزدیک به تبدیل KLT است [۱]. یک مزیت دیگر آن این است که الگوریتمهای سریع و موثری برای محاسبه آن وجود دارد که برای کاربردهای مبتنی بر پیاده‌سازی نرم‌افزاری مناسب است [۲].

از آنجا که پیکسلها هم در دو جهت مکانی با هم تزايد دارند و هم در بعد زمانی، بنابراین باید از تبدیل DCT سه بعدی استفاده کرد اما به دلیل مسایل مربوط به نیاز به حجم حافظه بالا و نیز تاخیر واحد رمزگذار کاربرد این نوع رمزگذاری در مخابرات محدود می‌گردد. بنابراین از تبدیل DCT دوبعدی استفاده می‌شود.

خود تبدیل DCT باعث فشرده‌سازی نمی‌شود بلکه استفاده از کوانتیزاسیون و کدگذاری VLC است که باعث تحقق فشرده‌سازی می‌گردند. تبدیل DCT به دلیل خاصیت متراکم‌سازی انرژی^{۱۷} به افزایش کارایی فشرده‌سازی کوانتایزر کمک می‌کند. در طراحی کوانتایزر از این ویژگی سیستم بینایی انسان استفاده می‌شود که به اعوجاجات ناشی از مولفه‌های فرکانس بالا چندان حساس نیست بنابراین مولفه‌های فرکانس بالای حاصل از تبدیل DCT را می‌توان با گام کوانتیزاسیون بزرگتری کوانتیزه کرد. در اثر این کار تعداد بیشتری از ضرایب صفر می‌شوند و در نتیجه میزان فشرده‌سازی بیشتر می‌شود گرچه کیفیت تصویر نیز تا حدی افت می‌کند.

۲-۲ کاهش تزايد زمانی

با استفاده از تفاضلات بین فریمهای متوالی می‌توان تزايد در حوزه زمان را کاهش داد. به این کار کدگذاری بین-فریمی گفته می‌شود. به ازاء آن نواحی‌ای از فریمها که ثابت بوده و هیچ تغییری در آن نواحی رخ نداده باشد، مقدار این تفاضل تقریباً صفر بوده و بنابراین نیازی به کدگذاری آنها نمی‌باشد. برای آن نواحی‌ای از فریمها که شامل حرکت اشیاء مختلف است اگر بتوان ابتدا بردار جابجایی را تخمین زده و تفاضل را نسبت به تصویر جبران شده محاسبه کرد باز هم می‌توان تفاضلات کوچک به دست آورده و به میزان فشرده‌سازی بالاتری دست یافت. بنابراین روشهای تخمین بردار جابجایی^{۱۸} در کدگذاری ویدیو اهمیت دارند.

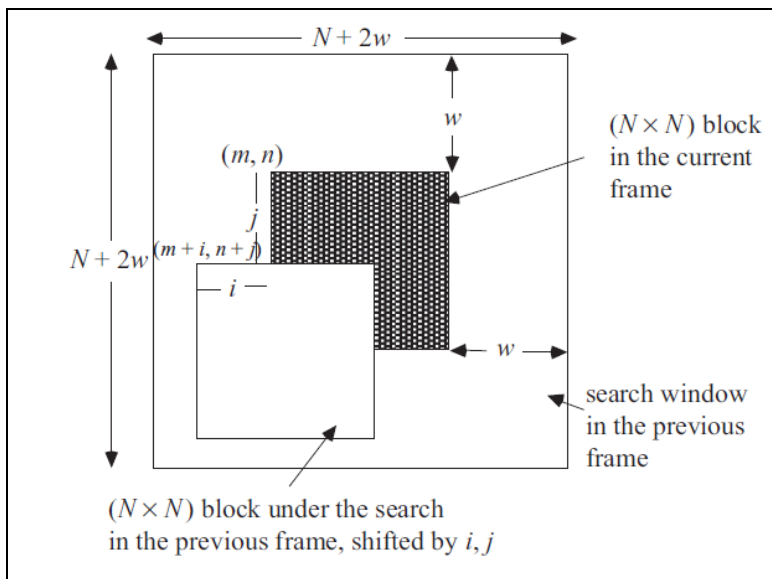
متداولترین روش تخمین حرکت (ME^{۱۹}) مورد استفاده در کدکهای استاندارد ویدیویی، الگوریتم انطباق بلوکی (BMA^{۲۰}) است. در نوع متداول از الگوریتم BMA هر فریم به بلوکهایی شامل $M \times N$ پیکسل (و یا بلوکهای

¹⁷Energy Compaction

¹⁸Motion Vector Estimation

¹⁹Motion Estimation

مربعی شامل N^2 پیکسل) تقسیم می‌شود [۳]. سپس هر بلوک متعلق به فریم فعلی، با تمام بلوکهای همسایه متناظر در فریم قبلی و واقع در یک پنجره جستجو (بر طبق یک معیار شباهت) مقایسه می‌شود تا شبیه‌ترین بلوک به آن شناسایی شود (شکل ۲-۳ ملاحظه شود). از آنجا بردار جابجایی مشخص می‌شود.



شکل ۲-۳ فریمهای فعلی و قبلی مورد استفاده در یک پنجره جستجو

برای بررسی انطباق دو بلوک از معیارهای مختلفی مانند تابع همبستگی متقابل (CCE)، میانگین مجذور خطا (MSE)، و میانگین قدرمطلق خطا (MAE) می‌توان استفاده کرد [۴-۶]. برای MSE:

$$M(i, j) = \frac{1}{N^2} \sum_{m=1}^N \sum_{n=1}^N (f(m, n) - g(m + i, n + j))^2, \quad -w \leq i, j \leq w \quad 1-2$$

برای معیار MAE:

$$M(i, j) = \frac{1}{N^2} \sum_{m=1}^N \sum_{n=1}^N |f(m, n) - g(m + i, n + j)|, \quad -w \leq i, j \leq w \quad 2-2$$

البته بارمحاسباتی معیار MAE بهتر از MSE است و بنابراین در تمام کدکهای ویدیویی استفاده می‌شود. حجم محاسباتی الگوریتم ME معمولاً حدود ۵۰ تا ۷۰ درصد از کل حجم محاسباتی الگوریتم کدگذاری را به خود اختصاص می‌دهد [۷]. بنابراین امروزه روشهای جستجو و محاسبه سریع بردارهای جابجایی زمینه‌ی فعالی از تحقیقات می‌باشد. ایده‌ی اصلی در اینگونه روشها این است که به جای استفاده از روش جستجوی کامل (FSM^{۲۱}) و بررسی تمام جابجاییهای ممکن، تنها تعداد محدودی از جابجاییهای ممکن در پنجره‌ی جستجو مورد بررسی قرار بگیرد. برای مثال در [۶] یک روش چند مرحله‌ای به نام TDL^{۲۲} پیشنهاد شده است که در هر مرحله تعداد محدودی جابجایی ممکن بررسی می‌شود و هر مرحله کار جستجو را در راستای بهترین جهت یافت شده در مرحله قبل ادامه می‌دهد. به علاوه در هر مرحله اندازه بازه جستجو یا گام حرکت نصف می‌شود. این کار تا زمانی که گام حرکت برابر ۱ پیکسل شود ادامه می‌یابد.

روش پیشنهادی در [۸] نیز تاحدی مشابه روش فوق‌الذکر است. در این روش که به نام روش سه مرحله‌ای (TSS^{۲۳}) شناخته شده است، تمام هشت جابجایی متعلق به پنجره جستجو به ابعاد w که با فاصله (یا گام حرکت جستجو) $w/2$ از هم و در اطراف این پنجره واقع‌اند ابتدا بررسی می‌گردند. در بهترین موقعیت (به نام موقعیت مینیمم)، اندازه گام حرکت جستجو نصف می‌شود و مجدداً هشت موقعیت مذکور نسبت به این موقعیت بررسی می‌شوند. این کار تا سه مرحله تکرار می‌شود تا تمام ۲۵ موقعیت ممکن بررسی شوند. روش TSS روش توصیه شده در پیاده‌سازی استاندارد H.261 است که در کاربردهای تلفنهای ویدیویی کاربرد دارد.

در روش [۴] (روش MMEA) قبل از نصف کردن گام حرکت دو موقعیت دیگر نیز علاوه بر موقعیتهای مذکور بررسی می‌شوند. در [۹] روشی به نام جستجوی ضربدری یا CSA^{۲۴} پیشنهاد شده که اساس آن مشابه روشهای فوق‌الذکر است یعنی از نصف کردن متوالی گام حرکت استفاده می‌کند اما به جای انتخاب بردارهای جابجایی گام یک کدک عمومی بین-فریمی را که در تمام استانداردهای ویدیویی مانند H.261، H.263، H.264،

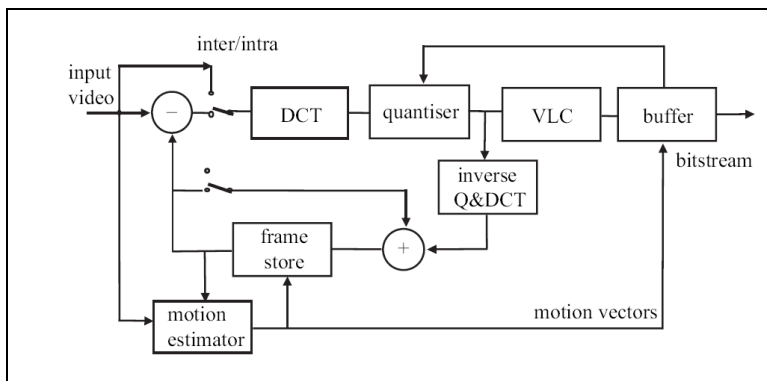
²¹Full Search Method

²²Two-Dimensional Logarithmic

²³Three Step Search

²⁴Cross Search Algorithm

MPEG-1، MPEG-2، و MPEG-4 مورد استفاده قرار می‌گیرد، نشان می‌دهد. هر یک از اجزای این بلوک دیاگرام در زیربخشهایی که در ادامه آمده‌اند، مختصراً توضیح داده می‌شوند.



شکل ۲-۶ یک کدگذار پیش‌بین بین فریمی عام

۲-۳-۱ حلقه‌ی بین-فریمی

در کدگذاری پیش‌بین بین-فریمی تفاضل بین مقادیر پیکسل‌های فریم فعلی و مقادیر پیش‌بینی شده‌ی آنها از طریق فریم قبلی کدگذاری و انتقال داده می‌شود. در واحد کدگشا، مقدار خطای کدگشایی شده مربوط به هر پیکسل به مقدار پیش‌بینی شده‌ی مشابهی اضافه شده تا مقدار اولیه پیکسل مذکور بازسازی گردد. هر چه از پیش‌بینی بهتری استفاده شود مقدار خطا کوچکتر و نرخ بیت نیز کمتر می‌شود. اگر صحنه‌ی ویدیو ثابت باشد بهترین تخمین از پیکسل فریم فعلی همان پیکسل متناظر از فریم قبلی است. اما اگر فرض کنیم که جابجایی انجام شده بین دو فریم از نوع انتقال ساده‌ی اشیاء باشد آنگاه از پیکسلی از فریم قبلی که با یک بردار جابجایی مناسب منتقل شده باشد، استفاده می‌گردد.

۲-۳-۲ تخمین‌گر حرکت

معمولاً تخمین حرکت برای هر پیکسل مستلزم انجام حجم بالای محاسبات است به همین دلیل تخمین و جبران حرکت به صورت گروهی برای پیکسل‌ها انجام می‌شود تا سربار محاسباتی به ازاء هر پیکسل کاهش یابد.

در کدکهای استاندارد معمولاً از بلوکهای به ابعاد 16×16 استفاده می‌شود. به چنین بلوکی یک ماکروبلوک²⁵ (MB) گفته می‌شود. در مورد تصاویر و فریمهای رنگی قابل توجه است که عمل تخمین حرکت فقط روی مولفه‌ی شدت روشنایی²⁶ انجام می‌شود. برای محاسبه‌ی بردار جابجایی در مورد مولفه‌های رنگی از همان بردار تخمین زده شده مذکور استفاده می‌شود.

۲-۳-۲ سوئیچ بین/درون

هر MB بسته به این که از نوع درون-فریمی باشد یا بین-فریمی به شیوه‌ی خاصی کدگذاری می‌شود. تعیین نوع هر MB نیز به شرایط مختلف و از جمله نوع استاندارد بستگی دارد. برای مثال، در استاندارد JPEG تمام MBها از نوع درون-فریمی در نظر گرفته می‌شوند.

۲-۳-۲ تبدیل DCT

هر MB به بلوکهای 8×8 شدت روشنایی و رنگ تقسیم شده و به واحد تبدیل DCT داده می‌شود. در هر MB چهار بلوک شدت روشنایی وجود دارد اما تعداد بلوکهای رنگ بستگی به درجه تفکیک رنگ (قالب تصویر) دارد.

۲-۳-۲ بافر ۵

نرخ بیت حاصل از خروجی کدگذار بین-فریمی متغیر بوده و در درجه‌ی اول بستگی به میزان فعالیت انجام شده در فریمها (یعنی حرکت اشیاء و جزئیات آنها) دارد. برای استفاده از این نرخ بیت متغیر در کانالهای با نرخ بیت ثابت، باید داده‌ها را در یک بافر ذخیره کنیم تا بتوانیم با نرخ بیت ثابت آنها را ارسال کنیم. هرگاه حجم داده‌های تولیدی آنقدر زیاد شود که بافر بهمرز اشباع و سرریز برسد، پسخوردی به واحد کوانتایزر ارسال می‌شود که باعث

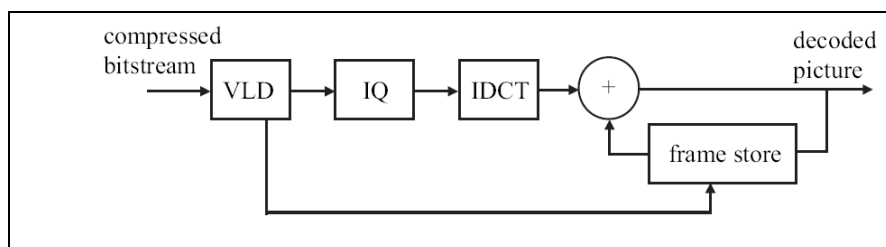
²⁵Macro Block

²⁶Luminance

افزایش گامهای مورد استفاده در کوانتایزر و در نتیجه کاهش نرخ بیت می‌شود. از سوی دیگر اگر نرخ داده‌های تولیدی کم شود پس‌خورد مذکور باعث کاهش گام کوانتایزر و در نتیجه افزایش نرخ بیت می‌شود.

۲-۳-۶ واحد رمزگشا

در واحد رمزگشا، پس از دی‌مالتی پلکس کردن و رمزگشایی با طول متغیر (VLD)، بردارهای جابجایی و ضرایب DCT از هم جدا شده و هرکدام در واحد مربوطه مورد استفاده قرار می‌گیرند. بردارهای جابجایی در واحد جبران حرکت و داده‌های DCT نیز پس از عبور از واحدهای عکس کوانتایزر و عکس DCT تبدیل به داده‌های خطا می‌شوند. سپس این داده‌های خطا به حاصل جبران حرکتی شده‌ی فریم قبلی اضافه می‌شوند تا فریم فعلی بازسازی شود. اجزاء واحد رمزگشا در شکل ۲-۷ نشان داده شده‌اند.



شکل ۲-۷ دیگرام بلوکی یک رمزگشا

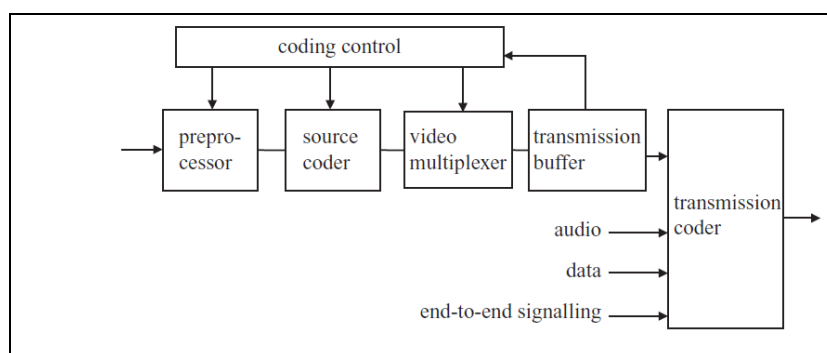
۲-۴ کدگذاری به منظور کنفرانس ویدیویی (H.261)

استاندارد H.261 به منظور ارائه روشهایی جهت کدگذاری، کدگشایی، و انتقال سیگنالهای ویدیویی روی شبکه‌های ISDN و با یکی از نرخهای انتقال $p \times 64$ kbit/s به وجود آمد. پارامتر p عددی بین ۱ تا ۳۰ می‌باشد. ساختار کدگذاری H.261 بسیار شبیه به دیگرام کدگذار عمومی نشان داده شده در شکل ۲-۶ می‌باشد. این کدگذار یک کدگذار بین-فریمی مبتنی بر DCT است. در این کدگذار ابتدا از پیش‌بینی بین-فریمی در حوزه پیکسلی استفاده می‌شود. در ادامه خطای پیش‌بینی به حوزه فرکانسی تبدیل شده و سپس به منظور کاهش پهنای باند از کوانتیزاسیون استفاده می‌گردد. در مرحله‌ی پیش‌بینی می‌توان از جبران حرکت (MC) نیز استفاده کرد گرچه این کار اختیاری است. بنابراین، در این کدگذار تزیاید زمانی از طریق پیش‌بینی بین-فریمی و

تزايد مکانی از طریق گرفتن تبدیل کاهش داده می‌شود. این نکته قابل ذکر است که در طراحی واحدهای رمزگذار و رمزگشا آزادی‌ها و اختیار عملهایی به طراح داده شده است اما به شرطی که همچنان از گرامر داده شده برای دنباله‌ی بیتی تبعیت کنند. گاهاً به جای نام H.261 از نام RM8 نیز استفاده می‌شود که اشاره به آخرین نسخه‌ی تصحیح شده (نسخه‌ی هشتم) از دستنوشته‌ی مرجع استاندارد^{۲۷} دارد که در سال ۱۹۸۹ تهیه گردیده است.

۲-۴-۱ قالب و ساختار ویدیو

شکل ۲-۸ بلوک دیاگرام مربوط به سیستم صوتی-تصویری مبتنی بر H.261 را نشان می‌دهد. در این بلوک دیاگرام واحد پیش‌پردازنده سیگنال ویدیویی ورودی را که به قالب CCIR-601 (قالب خروجی یک دوربین) است به قالب جدیدی تبدیل می‌کند. پارامترهای مختلف کدگذاری با دیگر داده‌ها (شامل داده‌های صوتی، داده‌های سیگنالینگ، و داده‌های سیگنال ویدیویی فشرده شده) مالتی پلکس و ترکیب می‌شوند. وظیفه بافر انتقال، کنترل نرخ بیت داده‌های ارسالی است.



شکل ۲-۸ دیاگرام بلوکی یک رمزگذار صوتی-تصویری H.261

در استاندارد H.261 امکان انجام درونبایی سه تصویر بین هر دو تصویر کدگذاری و ارسال شده وجود دارد. این کار باعث کاهش نرخ فریم از مقدار ۳۰ فریم بر ثانیه به مقادیر به ترتیب ۱۵ (اگر فقط یک تصویر درونبایی

²⁷Reference Manual (RM)

شود)، ۱۰ (اگر دو تصویر درونیایی شود)، و ۷.۵ فریم بر ثانیه (اگر سه تصویر درونیایی شود) می‌گردد. برای کاهش بیشتر نرخ بیت داده‌های ارسالی از قالب تصویری QCIF برای کاهش درجه تفکیک فریمها استفاده می‌شود.

در قالبهای تصویری CIF و QCIF بلوکهای تصویر به گروههایی به نام ماکروبلوک (MB) گروهبندی می‌شوند که هر ماکروبلوک شامل چهار بلوک شدت روشنایی^{۲۸} و دو بلوک رنگ^{۲۹} متناظر با هم می‌باشد. ماکروبلوکها به نوبه خود به لایه‌هایی به نام گروه بلوکها یا GOB ^{۳۰} تقسیم‌بندی می‌شوند. همان طور که در شکل ۲-۹ نشان داده شده است هر فریم به قالب CIF شامل ۱۲ عدد GOB و هر فریم به قالب QCIF شامل ۳ عدد GOB می‌باشد.

هدف از ساختار بندی فریمها بر حسب ماکروبلوکها و GOB ها عبارت است از:

✓ استفاده از کدگذاریهای مشابه در حالت‌های بین/درون-فریمی برای بلوکهای رنگ و شدت روشنایی

مربوط به ناحیه‌ی مشترک

✓ استفاده از یک بردار حرکت برای هر دو بلوک رنگ و شدت روشنایی

✓ کدگذاری موثر بلوکهای 8×8 تبدیل DCT

✓ کمک به از سرگیری ارسال داده‌ها در هنگام خرابی بیت‌های ارسالی

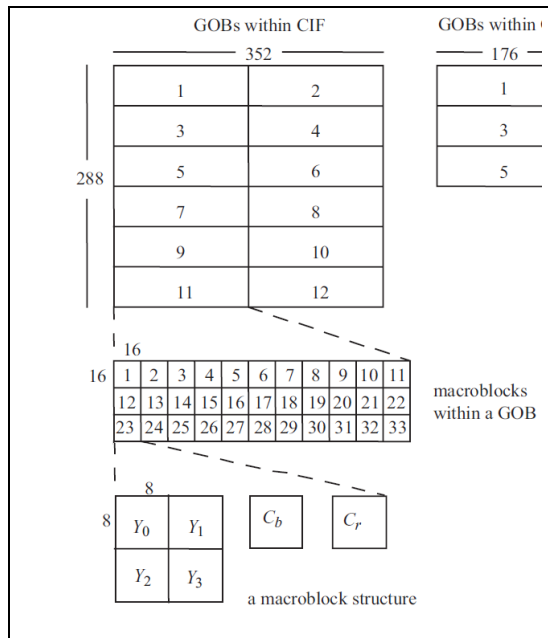
✓ حمل اطلاعات جانبی مربوط به GOB ، MB یا لایه‌های بالاتر. این اطلاعات شامل قالب تصویر، مراجع

زمانی، نوع MB ، اندیس کوانتایزر و غیره می‌شود.

²⁸Luminance

²⁹Chrominance

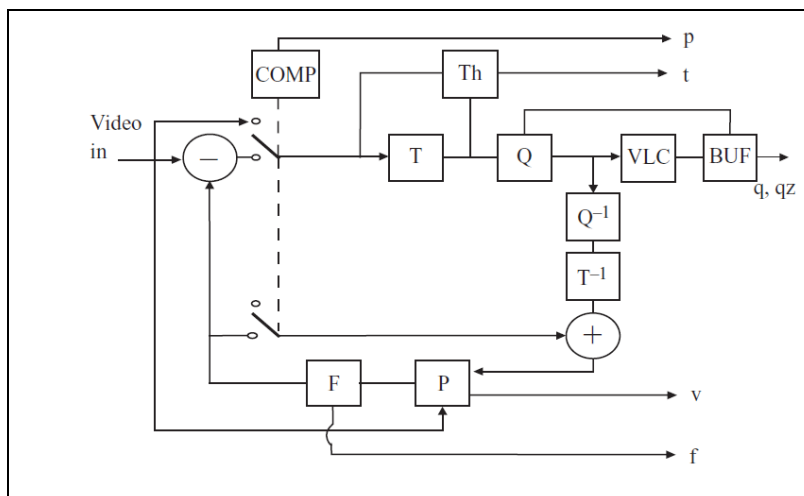
³⁰Group of Blocks



شکل ۲-۹ مفاهیم بلوک، ماکروبلوک و GOB در تصاویری با قالب CIF و QCIF

۲-۴-۲ الگوریتم کدگذاری ویدیو

بلوک دیاگرام الگوریتم کدگذاری در شکل ۲-۱۰ نشان داده شده است. این بلوک دیاگرام مشابه با بلوک دیاگرام مربوط به کدگذار عام (شکل ۲-۶) است. مهمترین عناصر این بلوک دیاگرام شامل پیش‌بین، جبران حرکت MC، تبدیل، کوانتیزاسیون، کدگذاری با طول متغیر VLC، و کنترل نرخ بیت می‌باشند.



شکل ۲-۱۰ دیاگرام بلوکی کدگذار ویدیویی H.261

خطای پیش‌بینی (در حالت بین-فریمی) و یا تصویر ورودی (در حالت درون-فریمی) به ماکروبلوکهای 16×16 تقسیم‌بندی می‌شوند که برخی از آنها ارسال خواهند شد. ماکروبلوکهایی که قرار است ارسال شوند به بلوکهای 8×8 تقسیم‌بندی شده و از مراحل تبدیل DCT، کوانتیزاسیون، و کدگذاری با طول متغیر عبور داده خواهند شد. معنای هر یک از واحدها و نمادهایی که در شکل ۲-۱۰ استفاده شده‌اند به صورت زیر است:

✓ COMP: یک مقایسه‌گر است که برای هر MB حالت کاری (درون/بین-فریمی) را تعیین می‌کند.

✓ Th: آستانه‌ای است که به منظور توسعه‌ی بازه‌ی کوانتیزاسیون استفاده می‌شود.

✓ T: تبدیل انجام شده روی بلوکهای 8×8

✓ T^{-1} : تبدیل معکوس

✓ Q: کوانتیزاسیون ضرایب تبدیل DCT

✓ Q^{-1} : عکس کوانتیزاسیون

✓ P: حافظه‌ی تصویر

✓ F: فیلتر حلقه

✓ p: پرچمی به منظور نشان دادن حالت‌های درون/بین-فریمی

✓ t: پرچمی به منظور نشان دادن انتقال دادن/ندادن

✓ q: اندیس کوانتیزاسیون برای ضرایب تبدیل

✓ v: اطلاعات مربوط به بردار جابجایی

✓ f: روشن و خاموش کردن فیلتر حلقه

۲-۴-۲-۱ پیش‌بینی

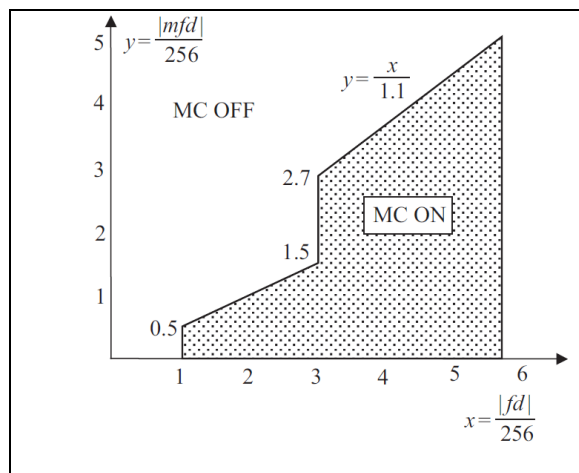
عمل پیش‌بینی به صورت بین-فریمی انجام شده و می‌تواند شامل جبران حرکت نیز باشد زیرا جبران حرکت (MC) در استاندارد H.261 اختیاری است. واحد رمزگشا به ازاء هر یک MB یک بردار جابجایی دریافت می‌کند.

مولفه‌های این بردار جابجایی اعداد صحیح بوده و از ± 15 پیکسل بر فریم فراتر نمی‌روند. تخمین حرکت فقط روی مولفه شدت روشنایی انجام می‌شود اما از همین بردار جابجایی محاسبه شده برای جبران حرکت همه‌ی چهار بلوک شدت روشنایی استفاده می‌شود. برای جبران حرکت دو بلوک رنگ، ابتدا بردار جابجایی مذکور را نصف و سپس به سمت صفر گرد می‌کنند.

برای انتقال بردارهای جابجایی محاسبه شده، تفاضل آنها از یکدیگر را محاسبه و از کدگذاری با طول متغیر استفاده می‌کنند.

۲-۲-۴-۲ تصمیم در مورد انجام جبران حرکت

تصمیم در مورد این که برای یک MB موجود نیاز به انجام جبران حرکت می‌باشد یا خیر بستگی به این دارد که آیا جبران حرکت باعث کاهش قابل توجه خطای پیش‌بینی می‌شود یا نه؟ برای انجام این تصمیم‌گیری از نمودار شکل ۲-۱۱ استفاده می‌شود. محور افقی نشان دهنده‌ی خطای پیش‌بینی نرمالیزه شده بدون جبران حرکت و محور عمودی نشان دهنده‌ی خطای پیش‌بینی نرمالیزه شده با انجام جبران حرکت می‌باشد. اگر مقادیر خطای پیش‌بینی با و بدون جبران حرکت به گونه‌ای باشند که در ناحیه‌ی هاشور زده شده قرار بگیریم، باید از جبران حرکت استفاده کنیم.



شکل ۲-۱۱ تصمیم‌گیری در مورد استفاده/عدم استفاده از جبران حرکت

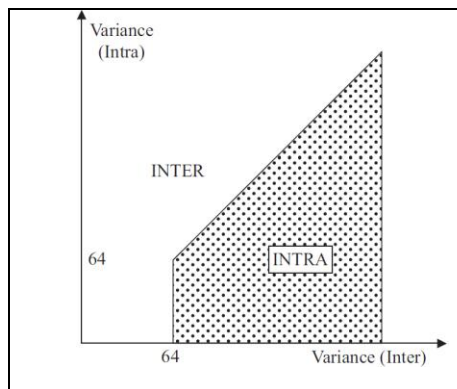
۳-۲-۴-۲ تصمیم‌گیری در مورد درون/بین-فریمی

گاهی اوقات استفاده از کدگذاری درون-فریمی در مورد یک MB حداقل به دودلیل زیر بهتر از کدگذاری بین-فریمی است:

- ۱- در حالتی که وقوع ناگهانی و قابل توجه حرکت داشته باشیم (مانند گذر از نوع برش^{۳۱}) خطای پیش‌بینی بین-فریمی کمتر از خطای پیش‌بینی درون-فریمی نخواهد بود.
- ۲- تصاویری که به روش درون-فریمی کدگذاری شوند پایداری خطای بهتری در مقابل خطاهای کانال از خود نشان می‌دهند. در کدگذاری بین-فریمی به این مساله توجه کنید که داده‌های دریافتی (توسط واحد رمزگشا) به فریم قبلی اضافه می‌شوند تا فریم فعلی را بازسازی کنند. در حالت وقوع خطاهای ناشی از کانال، مقادیر خطا در طول فریمهای متوالی منتشر خواهد شد و اثر نامطلوب روی آنها برجای خواهد گذارد.

در اینجا هم مشابه با حالت تصمیم‌گیری در مورد انجام یا عدم انجام جبران حرکت، از نمودار شکل ۲-۱۲ کمک می‌گیریم تا تعیین کنیم که برای MB موجود آیا باید از حالت بین-فریمی استفاده کنیم یا درون-فریمی. در اینجا محور افقی نشان‌دهنده‌ی واریانس MB در حالت استفاده از کدگذاری بین-فریمی و محور عمودی نشان‌دهنده واریانس MB در حالت استفاده از کدگذاری درون-فریمی است.

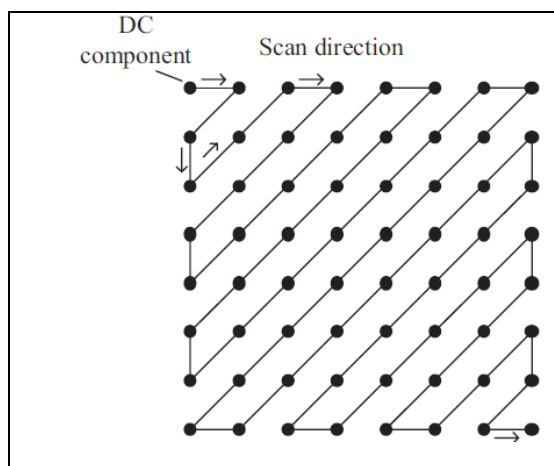
³¹Cut



شکل ۲-۱۲ تصمیم‌گیری در مورد انتخاب حالت‌های بین‌ادرون/فرون فریمی

۲-۴-۴-۲ کدگذاری با طول متغیر دوبعدی

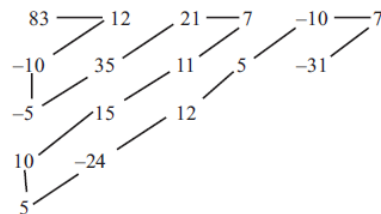
برای افزایش کارایی کدگذاری، بعد از انجام تبدیل روی بلوکهای 8×8 ، ابتدا به روش زیگزاگ این ضرایب جاروب شده و به صورت سطری مرتب می‌شوند.



شکل ۲-۱۳: نحوه جاروب ضرایب تبدیل 8×8

سپس ضرایب کوانتیزه شده و به روش کدگذاری با طول متغیر دوبعدی (2D-VLC) کدگذاری می‌شوند. برای

مثال [۱۱]:



Raw coefficients	83	12	-10	-5	35	21	7	11	15	10	5	-24	12	5	-10	7	-31
New coefficients	83	0	0	0	35	21	0	0	0	0	0	-24	0	0	0	0	-31
Quantised values	88	0	0	0	40	24	0	0	0	0	0	-24	0	0	0	0	-24
Index	5	0	0	0	2	1	0	0	0	0	0	-1	0	0	0	0	-1
Events to be transmitted: (run, index) (0,5) (3,2) (0,1) (5,-1) (4,-1)																	

در مثال فوق، به جای کدگذاری ۱۷ ضریب کوانتیزه شده، فقط کافی است ۵ جفت عدد (هر یک به نام واقعه^{۳۲}) کدگذاری و ارسال گردند.

۲-۴-۲-۵ حلقه فیلتر

در نرخ بیت‌های پایین اندازه گام کوانتایزر بزرگ شده و تعداد زیادی از ضرایب تبدیل DCT را صفر می‌کند. اگر تنها یک ضریب DC و تعداد محدودی ضریب AC وجود داشته باشد، تصویر بازسازی شده حالت بلوکی به خود می‌گیرد. در حالتی که با گذر فریم‌ها، محل نواحی بلوکی تغییر کند اثر آن مانند یک نویز فرکانس بالا مشاهده خواهد شد که به آن نویز mosquito گفته می‌شود. همچنین، اعوجاج ناشی از اثر بلوکی شدن تصویر در نواحی نزدیک به لبه‌های اریب مانند نویز پلکانی دیده می‌شود. برای مثال، فریم‌های زیر مربوط به نرخ ارسال kbit/s ۲۵۶ می‌باشند که دارای کیفیت کاملاً مناسب هستند.

³²Event



الف

ب

شکل ۲-۱۴ تصویری با نام Calire (الف) اصلی، (ب) کدگذاری شده با H.261 و در نرخ ۲۵۶ kbit/sec

اما در نرخ بیت‌های پایین‌تر اعوجاج قابل مشاهده می‌شود. فریم‌های زیر متناظر با فریم‌های فوق اما در نرخ بیت ۲۵۶ kb/s می‌باشند. در این حالت در اثر حرکت فریم‌ها با نرخ ۳۰ فریم بر ثانیه اثر نویز mosquito مشاهده خواهد شد.



الف

ب

شکل ۲-۱۵ تصویر Claire کدگذاری شده توسط H.261 و با نرخ (الف) ۱۲۸ kb/s، (ب) ۶۴ kb/s

در اثر انجام کوانتیزاسیون مولفه‌های فرکانس بالا حذف می‌شوند لذا آن را می‌توان معادل با یک فیلتر پایین‌گذر مدل کرد [۱۲ و ۱۳]. برای کاهش این اثرات منفی می‌توان از فیلتر حلقه استفاده کرد. این فیلتر، یک فیلتر

پایین گذر است که مولفه‌های فرکانس بالا را حذف کرده و اثر بلوکی شدن را تا حدی از بین می‌برد گرچه تا حدی اثر مات کردن نیز از خود ایجاد می‌کند. نتیجه اعمال فیلتر حلقه روی فریمهای شکل قبل در شکل ۲-۱۶ داده شده است:



ب

الف

شکل ۲-۱۶ تصویر کدگذاری شده با استفاده از فیلتر حلقه در حالت‌های (الف) ۱۲۸ kb/s، (ب) ۶۴ kb/s

فیلتر حلقه فقط در مورد بلوک‌هایی اعمال می‌شود که شامل حرکت باشند زیرا اگر آن را برای بلوک‌های ثابت اعمال کنیم اثر مات کردن آن، اثرات منفی تولید می‌کند. برای تحقق فیلتر حلقه معمولاً از پاسخ ضربه زیر استفاده می‌شود:

$$h(x, y) = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad ۳-۲$$

۲-۴-۳ کنترل نرخ

به دلیل متغیر بودن ماهیت و محتوای ویدئو، نرخ بیت خروجی کدگذار نیز متغیر است. برای ثابت کردن این نرخ از یک بافر استفاده می‌شود. به هنگام نزدیک شدن به حالت پرشدن بافر، اندازه گام کوانتایزر بزرگ و به

هنگام نزدیک شدن به حالت خالی شدن بافر، اندازه گام کوانتایزر کوچک می‌شود. معمولاً از رابطه‌ی ۴-۲ برای محاسبه اندازه‌ی گام کوانتیزاسیون استفاده می‌شود:

$$q = 2 \left\lfloor \frac{\text{buffer content}}{200p} \right\rfloor + 2 \quad 4-2$$

در رابطه‌ی فوق ملاحظه می‌شود که اندازه گام کوانتیزاسیون متناسب با محتوای بافر تغییر می‌کند. در رابطه فوق، پارامتر p ضریبی است که تعیین کننده اندازه نرخ بیت خروجی به صورت $p \times 64 \text{ kbit/s}$ می‌باشد. علامت $\lfloor \cdot \rfloor$ نیز معرف گرد کردن به سمت صفر است.

۲-۵ کدگذاری به منظور ذخیره اطلاعات ویدیویی دیجیتال (MPEG-1)

استاندارد MPEG-1 به منظور ذخیره و بازیابی اطلاعات ویدیویی دیجیتال پیشنهاد گردید. همچنین از این استاندارد انتظار می‌رفت که بتواند قابلیت‌هایی نظیر پخش رو به جلو^{۳۳}، ثابت کردن یک فریم، حرکت سریع به جلو^{۳۴}، حرکت معکوس سریع^{۳۵}، حرکت آهسته به جلو^{۳۶}، و دستیابی تصادفی^{۳۷} را پشتیبانی کند. معمولاً از این استاندارد برای رسیدن به نرخ بیت‌هایی در بازه 1-1.5 Mbit/s استفاده می‌شود.

هر دو استاندارد MPEG-1 و H.261 به منظور نرخ بیت‌های پایین و تصاویر با درجه تفکیک مکانی پایین طراحی شده‌اند. در MPEG-1 نیز از تبدیل DCT برای کدگذاری با تلافی خطاهای پیش‌بینی در حالت‌های درون/بین-فریمی استفاده می‌شود. استاندارد MPEG-1 تا حد زیادی نسخه‌ای توسعه یافته از H.261 محسوب شده و وجوه اشتراک زیادی با آن دارد. البته دنباله‌های بی‌تی^{۳۸} آنها متفاوت از یکدیگر است. در استاندارد MPEG-1 جزئیات

³³Forward Play

³⁴Fast Forward

³⁵Fast Reverse

³⁶Slow Forward

³⁷Random Access

³⁸Bitstream

طراحی رمزگشا بیان نشده است حتی در مورد رمزگذار نیز اطلاعات چندان زیادی داده نشده است. به همین دلیل طراحان از آزادی عمل زیادی برخوردار هستند.

۲-۵-۱ دوباره چینی تصویر

به دلیل وجود انتظار از استاندارد MPEG-1 در برقراری دو خواسته تاحدی متضاد، یعنی امکان دستیابی تصادفی به فریم دلخواه و امکان کدگذاری با کارایی بالا، گروه MPEG پیشنهاد داد که نباید تمام فریمها را به طور مشابه با هم کدگذاری کرد. آنها در یک دنباله‌ی ویدیویی چهار نوع تصویر پیشنهاد دادند. اولین نوع از تصاویر به نام تصاویر I^{۳۹} نامیده شده‌اند. اینها شامل تصاویری می‌باشند که به طور مستقل و بدون ارجاع به تصویر دیگری کدگذاری می‌گردند. از این تصاویر به عنوان نقاط دسترسی به نواحی مختلف ویدئو در مرحله‌ی رمزگشایی استفاده می‌شود (برقراری امکان دسترسی تصادفی به نواحی دلخواه از ویدئو). این تصاویر مشابه با استاندارد JPEG و با میزان فشرده‌سازی متوسط کدگذاری می‌شوند. دومین نوع، تصاویر نوع P^{۴۰} هستند. این تصاویر به طریق کدگذاری پیش‌بین و نسبت به تصویر I یا P کدگذاری شده‌ی قبلی کدگذاری می‌شوند. اینها خود به عنوان مرجع برای کدگذاری فریمهای بعدی مورد استفاده قرار می‌گیرند. کدگذاری این تصاویر بسیار مشابه با استاندارد H.261 است. سومین نوع از تصاویر، تصاویر B^{۴۱} یا تصاویر دوطرفه-کدگذاری شده می‌باشند به این معنا که برای کدگذاری آنها ممکن است از تصاویر قبلی، بعدی، و یا ترکیبی از آنها استفاده شود. این کار باعث افزایش کارایی جبران حرکت می‌شود زیرا برای جبران بخشهای روی هم افتاده از اشیاء مختلف بهتر عمل می‌کند. از تصاویر B هیچگاه برای پیش‌بینی استفاده نمی‌شود. این ویژگی ویژه MPEG-1 بوده و به دو دلیل زیر انجام می‌شود:

³⁹Independent

⁴⁰Predictive

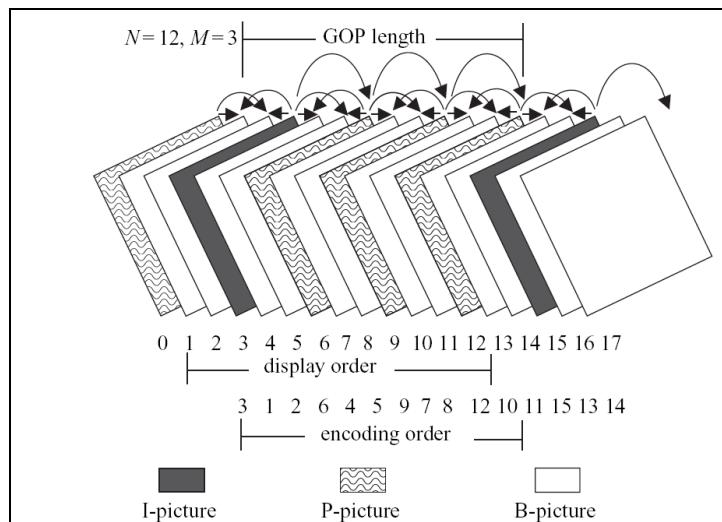
⁴¹Bidirectional

۱- اگر از تصاویر B برای پیش‌بینی فریم‌های بعدی استفاده نکنیم آن گاه آنها را می‌توان با بالاترین میزان فشرده‌سازی و بدون هیچ اثر جانبی منفی کدگذاری کرد. زیرا همان طور که می‌دانیم اگر یک فریم را با میزان فشرده‌ازی بالایی کدگذاری کنیم و از آن در پیش‌بینی فریم‌های بعدی استفاده کنیم آن گاه اعوجاجات ناشی از کدگذاری روی فریم بعدی تسری و روی آن اثر منفی خواهد داشت. بنابراین نیاز به صرف بیت‌های بیشتر برای حذف اعوجاجات خواهیم داشت.

۲- در کاربردهایی مانند انتقال ویدیو روی شبکه‌های مبتنی بر بسته^{۴۲} می‌توان به دلایل مختلف (مانند وقوع سرریز بافر) تصاویر B را حذف کرد بدون این که اثر منفی روی فریم‌های کدگذاری شده‌ی بعدی بگذارد. توجه به این نکته داریم که در تصاویر H.261 و یا تصاویر P و I (در MPEG-1) وقوع هر گونه خرابی یک فریم در حین انتقال، روی فریم‌های بعدی اثر منفی خواهد گذاشت تا زمانی که فریم از نو (و بدون استفاده از فریم‌های قبلی) بازسازی شود.

شکل ۲-۱۷ رابطه‌ی بین سه نوع فریم در MPEG-1 را نشان می‌دهد. از آنجا که تصاویر B برای پیش‌بینی از فریم‌های P و یا I قبل و بعد خود استفاده می‌کنند بنابراین باید بعد از کدگذاری این فریم‌ها، کدگذاری شوند. بنابراین ترتیب ورود فریم‌ها باید تغییر کند. این کار در واحد پیش‌پردازش انجام می‌شود.

⁴²Packet



شکل ۲-۱۷ مثالی از یک GOP در MPEG-1

چهارمین نوع تصویر در MPEG-1 تصویر D است. این نوع تصاویر به روش درون-فریمی کدگذاری شده و در کدگذاری آنها فقط از ضرایب DC استفاده می‌شود؛ بنابراین کیفیت آنها بسیار پایین است. از این فریمها معمولاً در کاربردهایی مانند پیش‌بردن سریع^{۴۳} استفاده می‌شود [۱۴ و ۱۵]. تصاویر D بخشی از یک GOP محسوب نمی‌شوند بنابراین آنها را در دنباله‌های شامل انواع دیگر تصاویر نمایش نمی‌دهند.

۲-۵-۲ ساختار ویدیو

۲-۵-۲-۱ گروه تصاویر

یک گروه از تصاویر یا GOP^{۴۴} گروهی شامل یک یا چند تصویر است که به منظور کمک به دسترسی تصادفی به دنباله‌ی تصاویر ایجاد شده‌اند. اولین تصویر کدگذاری شده در این گروه از نوع I بوده و به دنبال آن تعدادی تصویر P و B می‌آیند. نمونه‌ای از یک GOP در شکل قبلی (شکل ۲-۱۷) نشان داده شده است. طول هر GOP (پارامتر N در تصویر قبلی) معمولاً برابر فاصله‌ی هر دو تصویر I متوالی از یکدیگر است. فاصله‌ی هر تصویر P از تصویر P یا بعدی/قبلی را با پارامتر M نمایش می‌دهیم. انتخاب طول GOP بستگی به کاربرد دارد. در

⁴³Fast Forward

⁴⁴Group of Pictures

کاربردهایی که مستلزم دستیابی تصادفی، پخش سریع رو به جلو، و پخش سریع یا عادی رو به عقب می‌باشند، طول GOP را کوتاه انتخاب می‌کنند. همچنین، GOP را گاهاً در ابتدای وقوع یک برش^{۴۵} و هر جایی که جبران حرکتی موثر واقع نشده باشد انتخاب می‌کنند.

ترتیب کدگذاری تصاویر با ترتیب وقوع و نمایش آنها در دنباله‌ی ویدیویی متفاوت است. زیرا تصاویر B باید بعد از تصاویر P/I قبلی و بعدی خود کدگذاری شوند. در شکل اخیر ترتیب کدگذاری و ترتیب نمایش تصاویر مشخص شده است. به خاطر دوباره‌چینی انجام شده برای کدگذاری تصاویر، تاخیری در حد چند فریم در واحد رمزگذار رخ خواهد داد. همین مقدار تاخیر در واحد رمزگشا نیز وجود خواهد داشت. بنابراین این تاخیر کاربردهای MPEG-1 در مخابرات را محدود خواهد کرد [۱۴ و ۱۵].

۲-۲-۵-۲ تصویر

قالب همه‌ی انواع تصاویر I و P و B از نوع SIF می‌باشد. در قالب SID-625 مولفه شدت روشنایی هر تصویر شامل ۲۸۸ سطر و ۳۶۰ ستون و با نرخ فریم ۲۵ فریم بر ثانیه است. مولفه‌های رنگ نیز شامل ۱۴۴ سطر و ۱۸۰ ستون و با همان نرخ فریم بر ثانیه می‌باشند. در قالب SID-525 مولفه شدت روشنایی هر تصویر شامل ۲۴۰ سطر و ۳۶۰ ستون و با نرخ فریم ۳۰ فریم بر ثانیه است. مولفه‌های رنگ نیز شامل ۱۲۰ سطر و ۱۸۰ ستون و با همان نرخ فریم بر ثانیه می‌باشند.

۲-۲-۵-۲ قطعه

هر تصویر به تعدادی قطعه^{۴۶} تقسیم می‌شود. هر قطعه شامل گروهی از ماکروبلوکها است. در استاندارد H.261 به چنین گروهی از ماکروبلوکها یک GOB اطلاق می‌گردد. دلیل تعریف و استفاده از مفهوم قطعه مشابه با دلیل مربوط به تعریف GOB در H.261 است؛ یعنی، ریست کردن کدگذاری با طول متغیر (VLC) به منظور جلوگیری

⁴⁵Cut

⁴⁶Slice

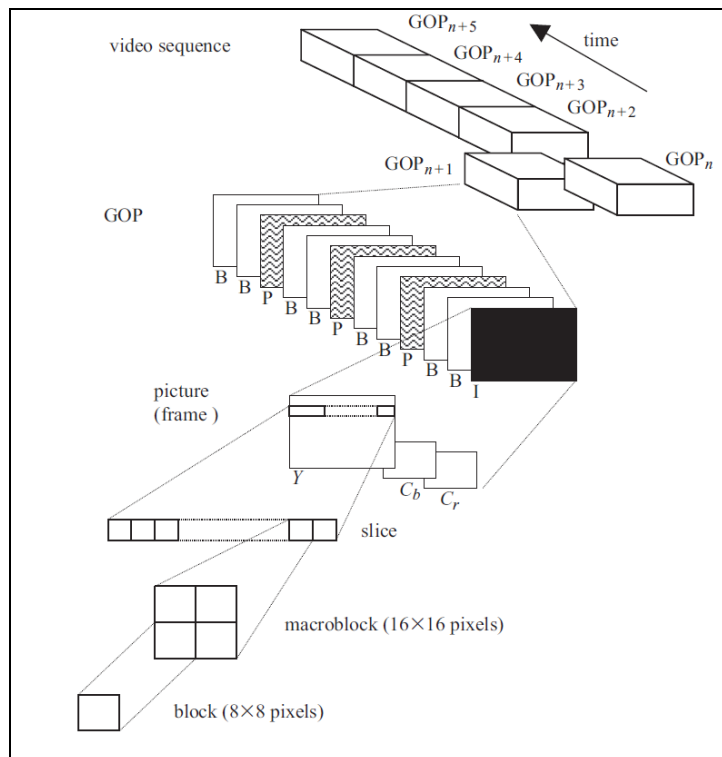
از نشت خطا در بخشهای مختلف یک تصویر. ابعاد قطعه‌ها داخل هر تصویر می‌تواند متغیر باشد و حتی از یک تصویر به تصویر دیگر نیز ابعاد آنها ممکن است متفاوت باشد.

۲-۵-۲-۴ ماکروبلوک

هر ماکروبلوک در حقیقت یک بلوک 16×16 از پیکسلها می‌باشد. ماکروبلوکها نیز به نوبه خود، برای کدگذاری، به تعدادی بلوک تقسیم می‌شوند. همان قوانینی که برای کدگذاری ماکروبلوکها در مورد H.261 بیان شد تقریباً در اینجا نیز صادق و برقرار است.

۲-۵-۲-۵ بلوک

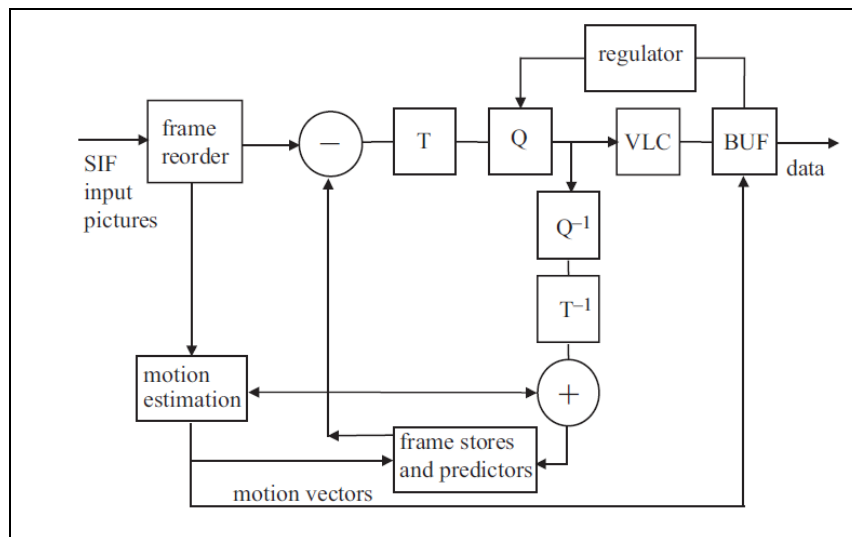
کوچکترین واحد هر تصویر یک بلوک است که شامل یک گروه 8×8 از پیکسلها (در مورد همه مولفه‌های شدت روشنایی و رنگ) می‌باشد. تبدیل DCT روی این بلوکها انجام می‌شود. شکل ۲-۱۸ ساختار یک دنباله‌ی ویدیویی را از بالا به پایین نشان می‌دهد. در بالاترین سطح، GOP، و در پایین‌ترین سطح بلوک قرار دارد.



شکل ۲-۱۸ ساختار کدگذاری ویدیو در MPEG-1

۲-۵-۳ واحد رمزگذار

همان طور که گفته شد استاندارد MPEG-1 نحوه طراحی واحدهای رمزگذار و رمزگشا را بیان نکرده است بلکه فقط ساختار دنباله‌ی بیتی را بیان کرده است. بنابراین بسیاری از گزینه‌ها مانند سرعت، نحوه برقراری مصالحه، کیفیت تصویر، و کارایی کدگذاری در اختیار طراح می‌باشد. با این حال شکل ۲-۱۹ یک راهنمای مناسب جهت طراحی واحد رمزگذار می‌باشد. همان طور که قبلاً نیز بیان شد این بلوک دیاگرام شبیه به بلوک دیاگرام رمزگذار عمومی و نیز رمزگذار استاندارد H.261 می‌باشد.



شکل ۲-۱۹ شمای ساده شده‌ی رمزگذار در MPEG-1

مهمترین تفاوت‌های بین رمزگذار فوق و رمزگذار H.261 شامل موارد زیر می‌باشند:

- ۱- دوباره‌چینی فریم: در ورودی رمزگذار فریم‌های B نگه داشته می‌شوند تا پس از اتمام کدگذاری فریم‌های P و یا B مربوطه، کدگذاری شوند.
- ۲- کوانتیزاسیون: ماکروبلوک‌هایی که به روش درون-فریمی کدگذاری شده‌اند، به منظور جبران اثر اعوجاجات کدگذاری، وزن‌دهی می‌گردند.
- ۳- تخمین حرکت: در اینجا نه تنها پنجره جستجو بزرگتر شده بلکه دقت جستجو نیز تا حد نصف یک پیکسل (از طریق درونیایی خطی) افزایش یافته است. برای فریم‌های B از جبران حرکت دوطرفه استفاده می‌شود.
- ۴- از هیچ فیلتر حلقه‌ای استفاده نشده است.
- ۵- استفاده از ذخیره‌سازی فریم و پیش‌بینیها به منظور پیش‌بینی فریم‌های B
- ۶- تنظیم کننده‌ی نرخ بیت: در MPEG-1 بیش از یک نوع تصویر وجود دارد که هرکدام نرخ بیت مخصوص به خود تولید می‌کنند. این مساله در تنظیم کننده نرخ در نظر گرفته شده است.

۲-۵-۴ ماتریس وزن دهی کوانتیزاسیون

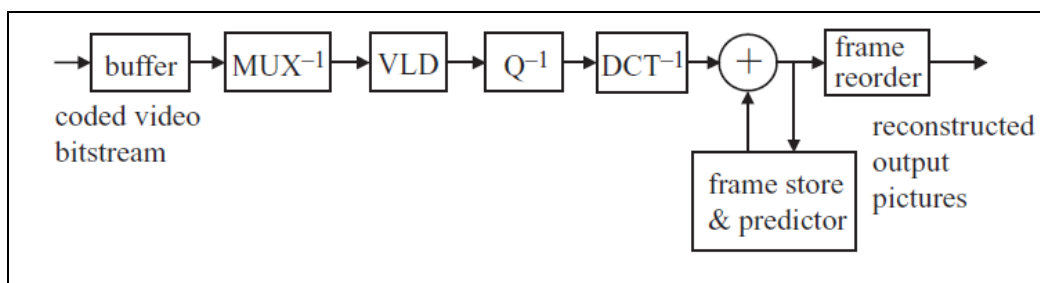
ضرایب تبدیل DCT قبل از کوانتیزاسیون تقسیم بر عناصر یک ماتریس به نام ماتریس وزن دهی کوانتیزاسیون می شوند تا از ویژگی سیستم بینایی انسان (عدم حساسیت نسبت به اعوجاجات فرکانس بالا) در جهت افزایش کارایی فشرده سازی استفاده شود. این ماتریس در MPEG-1 معمولاً به صورت شکل ۲-۲۰ انتخاب می شود.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

شکل ۲-۲۰ ماتریس کوانتیزاسیون

۲-۵-۵ واحد رمزگشا

دیاگرام بلوکی مربوط به یک نمونه رمزگشا در MPEG-1 در شکل زیر نشان داده شده است.



شکل ۲-۲۱ دیاگرام بلوکی رمزگشای MPEG-1

دنباله بیتی ورودی ذخیره و سپس دی مالتی پلکس می شود تا اجزای مختلف مانند ضرایب DCT، بردارهای جابجایی، انواع ماکروبلوکها، و آدرسها جدا شوند. سپس از رمزگشا با طول متغیر (VLD) به کمک جداول محلی

موجود استفاده می‌شود تا مقادیر عددی پارامترهای مختلف به دست آید. ضریب تبدیل DCT به دست آمده پس از عکس عمل کوانتیزاسیون، تبدیل معکوس DCT گرفته و به پیش‌بینی انجام شده به کمک جبران حرکت اضافه می‌شوند تا تصاویر بازسازی شوند. تصاویر بازسازی شده دوباره چینی می‌شوند تا به ترتیب مناسب برای نمایش قرار گیرند.

۲-۶ کدگذاری اطلاعات ویدیویی دیجیتال با کیفیت بالا (MPEG-2)

به دلیل شباهت عملکرد و موفقیت استانداردهای H.261 و MPEG-1 سازمانهای ISO/IEC و ITU-T تصمیم به انجام یک کار مشترک و تدوین یک استاندارد جدید گرفتند. این کار در سال ۱۹۹۵ به نتیجه رسید و استاندارد به نام MPEG-2 و یا H.262 به وجود آمد. البته این استاندارد عموماً به نام MPEG-2 شناخته می‌شود.

برخی کاربردهایی که استاندارد MPEG-2 برای آنها طراحی شده است عبارتند از:

- سرویسهای ماهواره‌ای باند وسیع (برای منازل)
- توزیع تلویزیون کابلی در شبکه‌های فیبر نوری و مسی و ...
- توزیع صدای دیجیتال از طریق کابل
- پخش صدای دیجیتال (ماهواره و تلویزیون)
- جمع‌آوری اخبار الکترونیکی
- سینمای الکترونیکی
- تئاتر تلویزیونی خانگی
- ارتباط بین اشخاص (مانند ویدئو کنفرانس و ویدئو فون)
- ذخیره‌ی تعاملی ویدئو (مانند CD)

در MPEG-1 مقاوم بودن نسبت به خطای انتقال چندان در نظر گرفته نشده بود اما در MPEG-2 این ویژگی در نظر گرفته شده است. در MPEG-2 دو نوع جریان بیتی تعریف شده است: جریان برنامه^{۴۷} و جریان انتقال^{۴۸}. جریان برنامه مشابه با MPEG-1 است گرچه از گرامر جدیدی استفاده شده تا بتواند قابلیت‌های جدید اضافه شده به MPEG-2 را پشتیبانی کند. در جریان انتقال، گرامر بسایر متفاوت از جریان برنامه بوده و قابلیت‌های مربوط به مقاوم‌سازی نسبت به کانال‌های نویزی به آن اضافه شده است [۱۶ و ۱۷].

۲-۶-۱ پروفایل‌ها و سطوح

استاندارد MPEG-2 به این منظور ایجاد گردید که حالتی عمومی داشته باشد به این معنا که محدوده وسیعی از کاربردهای مختلف، نرخ بیت‌های مختلف، درجات تفکیک مختلف، کیفیت‌های مختلف، و خدمات مختلف را پشتیبانی کند. برای این که از نظر پیاده‌سازی امکان تحقق اهداف وسیع فوق وجود داشته باشد، در استاندارد MPEG-2 از مفاهیم پروفایل^{۴۹} و سطح^{۵۰} به منظور ایجاد گروهی از گرامرهای^{۵۱} ممکن در دنباله‌ی بیتی مربوطه استفاده شده است [۱۶].

هر پروفایل یک زیرمجموعه از گرامر کلی دنباله بیتی تعریف شده در MPEG-2 است و منظور از هر سطح یک مجموعه‌ی مشخص از محدودیت‌های اعمال شده روی پارامترهای مورد استفاده در دنباله بیتی است. این محدودیت‌ها می‌توانند به سادگی شامل اعمال یک محدودیت روی مقادیر ممکن پارامترها باشند. سطوح و پروفایلها یک رابطه سلسله‌مراتبی با یکدیگر دارند به این معنا که هر گرامر تعریف شده توسط یک پروفایل یا سطح بالاتر باید تمام عناصر گرامری سطوح و پروفایل‌های پایین‌تر از خود را پشتیبانی کند.

⁴⁷Programme Stream

⁴⁸Transport Stream

⁴⁹Profile

⁵⁰Level

⁵¹Syntax

به منظور ارائه دنباله‌های بیتی به کاربردهای مختلف، این دنباله‌ها به صورت لایه لایه مرتب و عرضه می‌گردند. اگر فقط از یک لایه استفاده شود، به داده‌های کدگذاری شده‌ی ویدیویی، یک دنباله‌ی بیتی ویدیویی مقیاس‌ناپذیر^{۵۲} اطلاق می‌شود اما اگر از دو یا تعداد بیشتر لایه استفاده شود به آن سلسله مراتب مقیاس پذیر^{۵۳} گفته می‌شود. در حالت مقیاس پذیر به اولین (یا بالاترین) لایه، لایه پایه^{۵۴} گفته می‌شود. لایه پایه همیشه به طور مستقل رمزگشایی می‌گردد. دیگر لایه‌ها (لایه‌های پایین‌تر)، لایه‌های بهسازی^{۵۵} نامیده شده و فقط به کمک لایه‌های بالاتر خود قابل رمزگشایی هستند.

پروفایل‌های تعریف شده در MPEG-2 در جدول ۲-۱ نشان داده شده‌اند.

جدول ۲-۲ پروفایل‌های مختلف تعریف شده در MPEG-2

Type	Supporting tools	Application
Simple	I- and P-pictures, 4:2:0 format; non-scalable	Currently not used
Main	Simple profile + B-pictures	Broadcast TV
SNR scalable	Main profile + SNR scalability	Currently not used
Spatial	SNR profile + spatial scalability	Currently not used
High 4:2:2	Spatial profile + 4:2:2 format IBIBIB... pictures, extension of main profile to high bit rates	Currently not used Studio postproduction; high-quality video for storage (VTR) and video distribution
Multiview	Main profile + temporal scalability	Several video streams; stereo presentation

سطوح تعریف شده در MPEG-2 با مسائل مرتبط با درجه تفکیک تصویر سروکار دارند مانند تعداد پیکسلها به ازاء هر خط، تعداد خطوط به ازاء هر فریم، تعداد فریمها بر ثانیه، و نرخ بیت. مناسبترین سطوح مورد استفاده در MPEG-2 در جدول ۲-۲ آورده شده‌اند.

⁵²Nonscalable Video Bitstream

⁵³Scalable Hierarchy

⁵⁴Base Layer

⁵⁵Enhancement Layer

جدول ۲-۳ سطوح تعریف شده برای هر پروفایل در MPEG-2

Level	Resolutions	Simple	Main	SNR	Spatial	High	4:2:2	Multiview
		I, P 4:2:0	I, P, B 4:2:0	I, P, B 4:2:0	I, P, B 4:2:0	I, P, B 4:2:0 4:2:2	I, P, B 4:2:0 4:2:2	I, P, B 4:2:0
Low	pel/line			352	352			352
	line/frame			288	288			288
	frame/s			30/25	30/15			30/25
	Mbit/s			4	4			8
Main	pel/line	720	720	720		720	720	720
	line/frame	576	576	576		576	512/608	576
	frame/s	30/25	30/25	30/25		30/25	30/25	30/25
	Mbit/s	15	15	15		20	50	25
High 1440	pel/line	1440		1440	1440	1440		
	line/frame	1152		1152	1152	1152		
	frame/s	60		60	60	60		
	Mbit/s	60		60	80	100		
High	pel/line	1920			1920	1920		1920
	line/frame	1152			1152	1152		1152
	frame/s	60			60	60		60
	Mbit/s	80			100	130		300

۲-۶-۲ تفاوت‌های استاندارد MPEG-2 با استاندارد MPEG-1

۲-۶-۲-۱ تفاوت‌های اصلی

در MPEG-2 درجه‌های متنوعی از تفکیک مکانی تصویر امکان استفاده دارد از کیفیت در حد نوع SIF (۲۵) یا $30 \times 288 \times 352$ تا کیفیت در حد نوع HDTV ($60 \times 1250 \times 1920$). همچنین اکثر این تصاویر از نوع یک‌درمیان شده^{۵۶} می‌باشند حال آن که در MPEG-1 تصاویر از نوع پیش‌رونده^{۵۷} می‌باشند [۱۱].

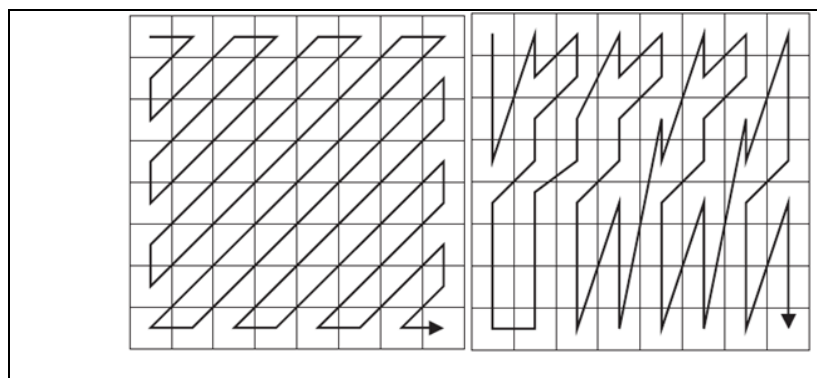
در MPEG-2 با در نظر گرفتن قالب‌های مختلف تصویری و نیز حالت‌های پیش‌رونده/یک در میان شده، محدوده‌ی جدیدی از انواع ماکروبلوکها (MB) ایجاد می‌شود.

۲-۶-۲-۲ تفاوت‌های فرعی

⁵⁶Interlaced

⁵⁷Progressive

اولین تفاوت مربوط به ترتیب جاروب ضرایب تبدیل DCT است. در MPEG-1 مشابه با H.261 از ترتیب جاروب زیگزاگ استفاده می‌شود اما در MPEG-2 علاوه بر جاروب زیگزاگ، امکان استفاده از جاروب متناوب^{۵۸} نیز وجود دارد.



ب

الف

شکل ۲-۲۱ نوع جاروب (الف) زیگزاگ، (ب) متناوب

تفاوت دوم مربوط به نحوه‌ی کوانتیزاسیون است. در MPEG-2، برخلاف MPEG-1، امکان استفاده از هر دو نوع کوانتیزاسیون خطی و غیرخطی وجود دارد. کوانتیزاسیون غیرخطی باعث افزایش کیفیت تصویر در نواحی با تابین پایین می‌گردد.

۲-۷ کدگذاری اطلاعات ویدیویی در کاربرد مخابرات نرخ بیت پایین (H.263)

استاندارد H.263 از نوعی نمایش رمزگذاری شده استفاده میکند که به منظور پشتیبانی از نرخ انتقالهای پایین در کاربردهایی نظیر شبکه‌های موبایل، شبکه‌های PSTN^{۵۹}، و شبکه‌های ISDN^{۶۰} طراحی شده است. نسخه آزمایشی این استاندارد در سال ۱۹۹۵ آماده گردید. این استاندارد، توسعه یافته‌ی استانداردهای H.261 و

⁵⁸Alternate

⁵⁹Public Switched Telephone Network

⁶⁰Integrated Services Digital Network

MPEG-1,2 است. برای این منظور ابعاد تصاویر مورد نظر کوچک انتخاب گردید. در استاندارد H.263 استفاده از پنج قالب sub-QCIF، QCIF، CIF، 4CIF، 16CIF (مطابق با جدول ۲-۳) توصیه گردیده است.

جدول ۲-۴ پنج قالب تصویری مورد پشتیبانی H.263

Picture format	Number of pixels for luminance per line	Number of lines for luminance per picture	Number of pixels for chrominance per line	Number of lines for chrominance per picture
Sub-QCIF	128	96	64	48
QCIF	176	144	88	72
CIF	352	288	176	144
4CIF	704	576	352	288
16CIF	1408	1152	704	576

البته بعد از ارائه H.263 تلاشهایی در جهت رسیدن به کاربردهای مخابرات بلادرنگ نیز صورت گرفت که منجر به پیشنهاد H.263+ گردید [۱۷]. این تلاشها ادامه یافت تا اینکه در سال ۲۰۰۰ نسخه توسعه یافته‌ی بعدی نیز با نام H.263++ نیز ارائه گردید [۱۸]. فعالیت دیگری که در سازمان استاندارد ITU-T به موازات صورت گرفت شامل همکاری با گروه مربوط به استاندارد MPEG-4 و تهیه استاندارد به نام [H.263L ۱۹] بود. در همه‌ی این استانداردها هدف اصلی رسیدن به مخابرات بلادرنگ بود که مستلزم رعایت دو محدودیت مهم زمان تاخیر و پیچیدگی واحدهای رمزگذار و رمزگشا است.

۲-۷-۱ تفاوت‌های H.263 با استانداردهای H.261 و MPEG-1

استاندارد H.263 از همان ساختار کلی کدگذاری عام (کدگذاری بین-فریمی مبتنی بر DCT) تبعیت می‌کند. مهمترین تفاوت بین استانداردهای H.263 و H.261 (و البته MPEG-1) در نحوه‌ی کدگذاری ضرایب تبدیل و بردارهای جابجایی است.

در استانداردهای H.261 و MPEG-1 ابتدا ضرایب تبدیل با جاروب زیگزاگ و توصیف دوره‌ی تداوم دوبعدی (2D-RLE) به پیشامدهایی دوبعدی به صورت (run,index) تبدیل می‌شدند که در آن منظور از run تعداد ضریبهای صفر قبل از ضریب غیرصفر فعلی و index شماره ضریب غیر صفر کوانتیزه شده در مجموعه گامهای کوانتایزر است. در استاندارد H.263 از پیشامدهای سه بعدی به صورت (last,run,level) استفاده می‌شود. پارامترهای run و level معادل همان پارامترهای run و index در استاندارد H.261 می‌باشند. پارامتر last فقط دو مقدار ممکن ۰ و ۱ می‌تواند بگیرد. اگر مقدار آن صفر باشد به این معنا است که هنوز به آخرین ضریب غیر صفر مربوط به بلوک فعلی نرسیده‌ایم اما اگر این پارامتر برابر ۱ باشد یعنی به آخرین ضریب غیرصفر رسیده‌ایم. در H.263 برای هر ماکروبلوک (به ابعاد ۱۶×۱۶) یک بردار جابجایی با دقت نصف پیکسل محاسبه می‌شود. سپس برای هر بردار جابجایی به کمک بردارهای محاسبه شده‌ی همسایه یک پیش‌بینی (یا تقریب) محاسبه شده و تفاضل آنها کدگذاری می‌گردد. برای محاسبه پیش‌بینی برای بردار جابجایی فعلی MV از بردارهای جابجایی نشان داده شده در شکل ۲-۲۲ و برای تقریب آن از روابط ۲-۵ و ۲-۶ استفاده می‌شود.

	$MV2$	$MV3$
$MV1$	MV	

MV : current motion vector
 $MV1$: previous motion vector
 $MV2$: above motion vector
 $MV3$: above right motion vector

شکل ۲-۲۲ پیش‌بینی بردار جابجایی

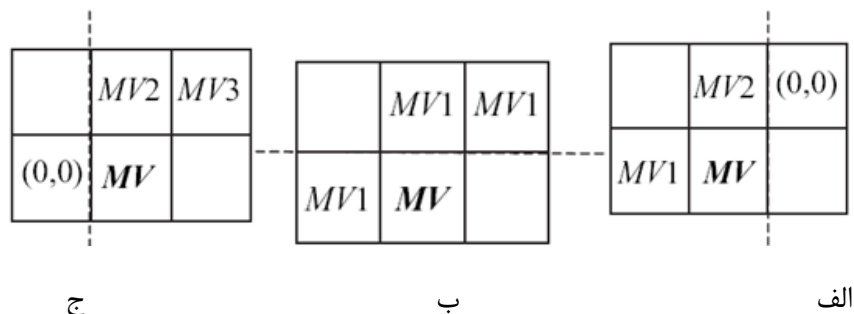
$$\begin{aligned} \text{pred}_x &= \text{median}(MV1_x, MV2_x, MV3_x) \\ \text{pred}_y &= \text{median}(MV1_y, MV2_y, MV3_y) \end{aligned}$$

۵-۲

$$\begin{aligned} MVD_x &= MV_x - \text{pred}_x \\ MVD_y &= MV_y - \text{pred}_y \end{aligned}$$

۶-۲

برای نواحی مرزی نیز که برخی ماکروبلوکهای همسایه وجود ندارند از حالت‌های نشان داده شده در شکل ۲-۲۳ استفاده می‌شود.



شکل ۲-۲۳ پیش‌بینی بردار جابجایی برای ماکروبلوکهای مرزی در حالت‌های مختلف. علامت خط‌چین معرف مرز تصویر است.

۲-۸ کدگذاری ویدئو مبتنی بر محتوا (MPEG-4)

استانداردهای MPEG-1 و MPEG-2 بیشتر روی طرح روشهای کدگذاری در کاربردهای به ترتیب CD-ROM و تلویزیون دیجیتال تمرکز دارند. در مقابل استاندارد MPEG-4 روی کاربردهای ذخیره‌سازی کارآمد، انتقال و دستکاری داده‌های ویدیویی در محیط‌های مالتی مدیا تمرکز یافته است [۲۱ و ۲۲].

روشی که در استاندارد MPEG-4 در پیش گرفته شده است مبتنی بر نمایش داده‌های بصری مبتنی بر محتوا است؛ به این معنا که برخلاف سایر استانداردهای بیان شده تا کنون، یک صحنه از ویدئو به صورت ترکیبی از اشیاء ویدیویی یا VO⁶¹ نگاه می‌شود که هر شیء دارای ویژگیهای ذاتی مانند شکل، حرکت و بافت است. این ایده کلید اصلی موفقیت استاندارد MPEG-4 در کاربردهای مبتنی بر رابطه تعاملی با مالتی مدیا است. در چنین کاربردهایی کاربر قدرت دسترسی به اشیاء موجود در ویدئو (با هر شکل دلخواه) و دستکاری آنها را دارد.

⁶¹Video Object

برای استاندارد MPEG-4 نیز یک کدک مبنا یا مدل تایید^{۶۲} (VM) طراحی شده است [۲۳] که هر از چند گاهی در آزمایشگاههای سرتاسر دنیا روی آن کار شده و نسخه بهبود یافته آن عرضه می‌گردد. این مدل تایید تاکنون از نسخه‌های 1.0 تا 11.0 تغییر و بهبود یافته است.

اگر تمام یک فریم را به عنوان یک شیء در نظر بگیریم آن گاه نوع نگرش استاندارد MPEG-4 با بقیه استانداردها یکسان می‌گردد. این حالت به نام MPEG-4 visual شناخته شده است. اما حالت معمولی و متداول آن که هر فریم را به اشیاء مختلف تجزیه می‌کند به نام MPEG-4 part 10 شناخته می‌شود که البته نام دیگر استاندارد H.264 است.

۲-۸-۱-۲ پروفایلها و سطوح

در MPEG-4 نیز مشابه با MPEG-2، قابلیتها و کاربردهای زیادی در نظر گرفته شده است که کاربران ممکن است تنها به قسمتی از این قابلیتها علاقه‌مند باشند. به این قابلیتها، پروفایل گفته می‌شود. برای هر پروفایل تعدادی محدودیت تعریف می‌شود (مانند نرخ بیت، نرخ فریم، و درجه تفکیک پیکسلی) که به آنها سطوح اطلاق می‌گردد.

۲-۸-۲-۲ صفحه اشیاء ویدیویی (VOP^{۶۳})

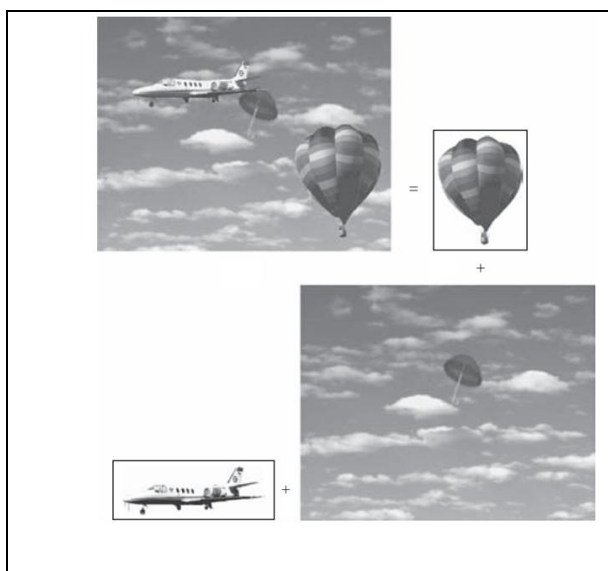
در MPEG-4 هر فریم به عنوان لایه‌هایی از صفحات اشیاء ویدیویی (VOP) تعریف می‌شود. هر VOP در حقیقت یک تصویر (یا فریم) شامل یکی از اشیاء موجود در ویدیو است. شکل ۲-۲۴ یک نمونه فریم ویدیویی را نشان می‌دهد که به سه VOP تجزیه شده است. در این مثال دو شیء مورد علاقه ما یعنی بالن و هواپیما از فریم اصلی

⁶²Verification Model

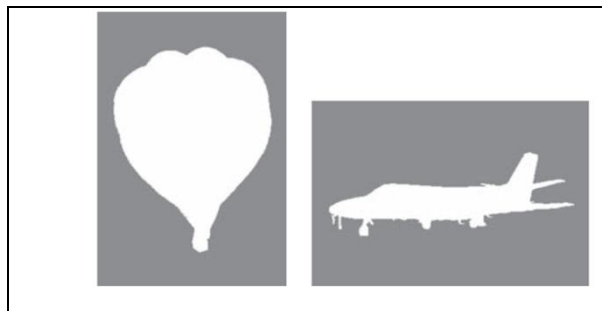
⁶³Video Object Plane

جدا شده و در صفحه‌های VOP_1 و VOP_2 قرار داده شده‌اند. باقیمانده‌ی فریم اصلی به عنوان پس‌زمینه در نظر گرفته شده و به نام VOP_0 نامگذاری شده است.

از دیدگاه کدگذاری، کافی است پس‌زمینه را تنها یکبار کدگذاری کنیم اما بقیه صفحات شیء را نسبت به محور زمان کدگذاری کنیم. از آنجا که در هر فریم واحد کدگذار تنها با اشیاء سروکار دارد و این اشیاء قسمت کوچکی از کل تصویر را به خود اختصاص داده‌اند، بنابراین میزان فشرده‌سازی تا حد بسیار زیادی افزایش می‌یابد. برای نمایش اشیاء دو راه وجود دارد که هر دو در MPEG-4 پشتیبانی شده است. راه اول استفاده از یک صفحه‌ی سطح خاکستری به نام صفحه آلفا و راه دوم استفاده از روشهای نیمه خودکار ناحیه‌بندی است. در روش اول، هر مقدار از صفحه آلفا یک عدد ۸ بیتی است که حامل اطلاعات مربوط به شکل اشیاء است. بنابراین می‌توان تا ۲۵۶ شیء مختلف را در این صفحه توصیف کرد. در روش دوم برای توصیف شکل هر شیء از یک تصویر دودویی به نام ماسک استفاده می‌شود. نمونه‌ای از این روش در شکل ۲-۲۵ نشان داده شده است که مربوط به مثال قبلی است.



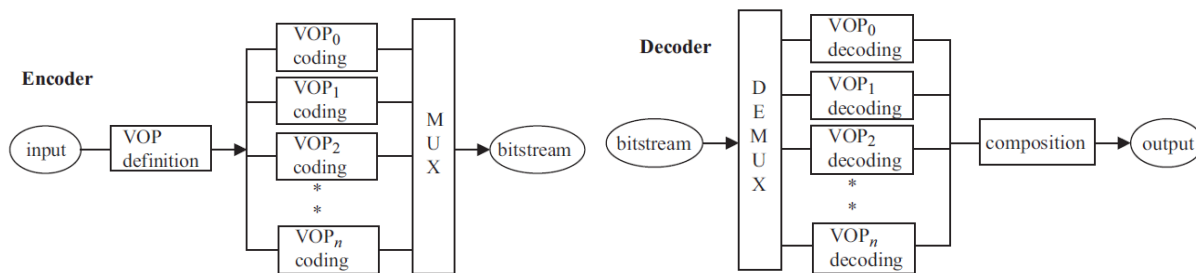
شکل ۲-۲۴ مثالی از صفحه اشیاء ویدیویی (VOP)



شکل ۲-۲۵ استفاده از ماسک جهت نمایش اشیاء

۲-۸-۳ کدگذاری اشیاء

واحد رمزگذار علاوه بر اینکه باید VOPها را کدگذاری کند اطلاعات مربوط به چگونگی ترکیب آنها با هم به منظور تشکیل فریم نهایی را نیز کدگذاری کند. برخی از این اطلاعات شامل مکان و زمان ترکیب VOP مورد نظر می‌باشند. به کاربران اجازه ردگیری شیء یا اشیاء مورد نظرشان داده شده است. آنها حتی می‌توانند نحوه ترکیب و تشکیل کل صحنه ویدیویی را تغییر دهند. بلوک دیاگرام کدک مورد استفاده در مدل تایید در شکل ۲-۲۶ نشان داده شده است.



ب

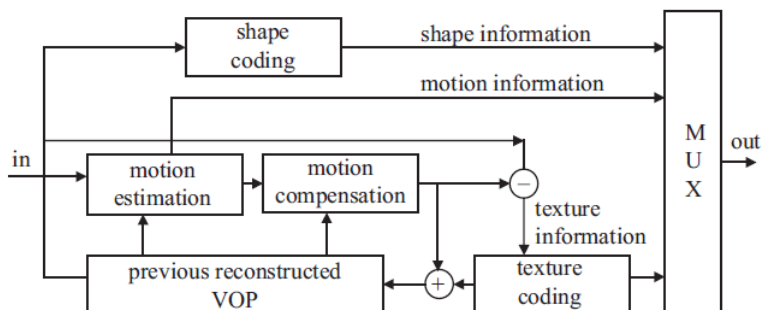
الف

شکل ۲-۲۶ دیاگرام بلوکی کدک مورد استفاده در مدل تایید، (الف) رمزگذار، (ب) رمزگشا

۲-۸-۳-۱ کدگذاری VOPها

شمای کلی و بلوک دیاگرامی ساختار کدگذار VOPها در شکل ۲-۲۷ نشان داده شده است. این کدگذار از دو بخش عمده تشکیل شده است: کدگذار شکل و یک نمونه مرسوم کدگذار حرکت و بافت (برای مثال H.263).

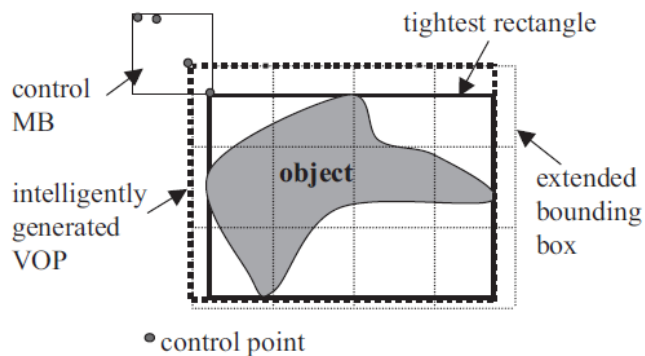
قبل از اینکه توضیح دهیم که بافت و شکل چگونه کدگذاری می‌شوند، ابتدا توضیح می‌دهیم که یک VOP چگونه باید نمایش داده شود.



شکل ۲-۲۷ ساختار کدگذار ویدیویی

۲-۳-۸-۲ تشکیل VOPها

برای تشکیل یک VOP از اطلاعات شکل آن استفاده می‌شود. برای پیشنهاد کردن کارایی کدگذاری، VOP با شکل دلخواه در یک مستطیل محیطی طوری قرار داده می‌شود که شیء شامل کمترین تعداد ماکروبلوکها شود. یک مثال در شکل ۲-۲۸ نشان داده شده است. ملاحظه می‌کنید که صفحه توسعه یافته‌ای نیز در حین محاسبه مستطیل محیطی محاسبه می‌شود.



شکل ۲-۲۸ مثالی از نحوه تشکیل یک VOP

۲-۸-۳ ناحیه بندی تصویر

اگر در دنباله‌ی ویدیویی، قبلاً اشیاء مختلف مشخص نشده باشند، باید به کمک ناحیه بندی برای هر فریم اشیاء مناسب را استخراج کرد.

۲-۸-۳-۴ کدگذاری شکل

منظور از شکل باینری و شکل سطح خاکستری، همان صفحه آلفای باینری و صفحه آلفای سطح خاکستری است. صفحات آلفای باینری به کمک یکی از روشهای کدگذاری شکلهای باینری کدگذاری می شود که بعداً توضیح داده می شوند. صفحات آلفای سطح خاکستری به کمک تبدیل DCT جبران حرکتی شده و مشابه با کدگذاری بافت، کدگذاری می شوند. به ماکروبلوکهای متعلق به صفحه آلفای توسعه یافته (که در بخش قبل محاسبه شدند)، بلوکهای آلفا می گوئیم. برای هر بلوک آلفا، یک بار فرآیند کدگذاری و کدگذاری تکرار می شود. برای کدگذاری یک صفحه آلفای باینری چندین روش پیشنهاد شده است که عبارتند از: توصیف مرزهای شیء به کمک کدزنجیره‌ای^{۶۴}، کدگذاری به روش درخت چهارتایی^{۶۵}، الگوریتم نی اصلاح شده (MMR)^{۶۶}، و کدگذاری حسابی مبتنی بر فحوا (CAE)^{۶۷} که از این بین به نظر می رسد روش CAE کارایی بهتری نسبت به دیگر روشها دارد.

کدگذاری صفحه آلفای سطح خاکستری مشابه با روش مورد استفاده برای کدگذاری مولفه شدت روشنایی است.

۲-۹ کدگذاری ویدیویی پیشرفته (H.264)

استاندارد H.264 محصول مشترک سازمانهای ITU-T و ISO/IEC (گروه کاری MPEG-4) است که در سال ۲۰۰۳ توسط سازمان ITU-T نهایی و اعلام گردید. این استاندارد در سازمان ISO به نام 14496-10 و یا

⁶⁴Chain Code

⁶⁵Quad Tree

⁶⁶Modified Modified Reed (MMR)

⁶⁷Context-based Arithmetic Encoding

MPEG4-part 10 شناخته شده است. البته این پروژه در شروع کار خود با نام AVC⁶⁸ شروع به کار کرده و به همین دلیل به طور غیررسمی به همین نام خوانده می شود. ما از این به بعد این استاندارد را با نام H.264/AVC و یا به سادگی H.264 استفاده می کنیم. کارایی فشرده سازی این استاندارد حداقل دوبرابر استاندارد MPEG-2 است به همین دلیل امروزه (در سال ۲۰۱۰) عمده ی کشورها از این استاندارد در کاربردهای متنوعی استفاده می کنند.

برخی ویژگیهای برجسته این کدک به قرار زیر است:

- کاهش نرخ بیت تا ۵۰ درصد نسبت به استانداردهای H.263+ و MPEG-4 visual
- پشتیبانی از حالت تاخیر کم برای سازگاری با برخی کاربردهای مخابراتی و پشتیبانی از حالت حجم پردازش بالا در کاربردهایی که تاخیر زمانی مساله مهمی برای آنها محسوب نمی شود مانند ذخیره سازی

ویدئو

- مقاوم بودن نسبت به خطای کانال
 - سازگاری و انعطاف با ساختار شبکه های انتقال داده
- برخی نقاط تمایز این استاندارد از نظر کاری به قرار زیر است [۲۴]:

- استفاده از دو کدگذاری آنروپی: کدگذاری با طول متغیر و وابسته به فحوا (CAVLC⁶⁹)
- انجام جبران حرکتی به کمک چندین تصویر مرجع
- انجام تخمین حرکت با بلوکهایی با ابعاد متغیر از ۴×۴ تا ۱۶×۱۶
- انجام جبران حرکت با دقت یک چهارم پیکسل، بهبود دقت پیش بینی، و استفاده از درونیایی با پیچیدگی محاسباتی کمتر
- استفاده از تبدیل معکوس پذیر عدد صحیح به منظور جلوگیری از عدم تطابق تبدیل و عکس آن

⁶⁸Advanced Video Coding

⁶⁹Context-Adaptive Variable Length Coding

- استفاده از بلوکهای با ابعاد متغیر در تبدیل عدد صحیح 8×8 و 4×4
- استفاده از فیلتر حذف اثر بلوکی^{۷۰} ایجاد شده به دلیل جبران سازی حرکت و کوانتیزاسیون
- مقاوم سازی بیشتر نسبت به خطا
- بخش بندی داده ها ($DP^{۷۱}$) و بسته بندی قطعه های ویدیویی به سه سطح اولویت به منظور محافظت در حین انتقال

- انتقال با تزیاید برخی نواحی به منظور بهبود مقاوم سازی نسبت به گم شدن داده ها

۲-۹-۱ قالب تصویر

استاندارد H.264 از هر دو قالب یک درمیان شده^{۷۲} و پیش رونده^{۷۳} به عنوان قالبهای ممکن فریمهای ویدیویی پشتیبانی می کند. مشابه با بقیه کدکهای استاندارد، در اینجا هم یک ماکروبلوک (MB) به ابعاد 16×16 بوده و به عنوان واحد پایه برای فرآیند فشرده سازی محسوب می شود.

۲-۹-۲ قطعه سازی^{۷۴}

به گروهی از ماکروبلوکها یک قطعه گفته می شود. هر تصویر متشکل از تعدادی قطعه است. قطعه ها مستقل از یکدیگر کد گذاری و کد گشایی می شوند. در H.264 شکل قطعه ها دلخواه بوده و متفاوت با بقیه استانداردهای ذکر شده تاکنون می باشد. نمونه ای از یک فریم به همراه قطعه های تعریف شده در آن در شکل ۲-۲۹ نشان داده شده است.

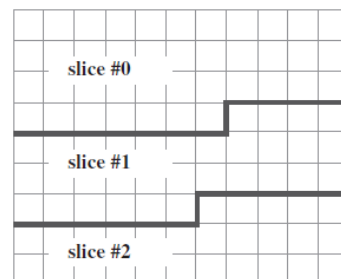
⁷⁰Deblocking Filter

⁷¹Data Partitioning

⁷²Interlaced

⁷³Progressive

⁷⁴Slicing



شکل ۲-۲۹ نمونه‌ای از یک قطعه

۲-۹-۱ انواع قطعه‌ها

در H.264 پنج نوع قطعه تعریف شده است [۲۴]:

- قطعه I: تمام ماکروبلوکهای آن به روش درون-فریمی کدگذاری می‌شوند.
- قطعه P: علاوه بر کدگذاری درون-فریمی، برخی MBها را میتوان به طریق بین-فریمی و استفاده از حداکثر یکبار پیش‌بینی جبران‌سازی حرکتی برای هر بلوک پیش‌بینی کرد.
- قطعه B

۲-۱۰ توصیف محتوا، جستجو و حمل (MPEG-7 و MPEG-21)

استاندارد MPEG-7 بیشتر روی موضوع توصیف کیفی ویدیو و جستجوی محتویات آن تمرکز یافته است [۲۵]. برای مثال، در کاربردهایی که با در دست داشتن یک تصویر کلیدی به دنبال جستجوی محتویات یک دنباله‌ی ویدیویی هستیم، این استاندارد مفید واقع می‌شود. استاندارد MPEG-21 به دنبال ارائه یک راهکار استاندارد برای تبدیل محتویات ویدیو به المانهای ویدیویی (با تعریف مشخص) به منظور فراهم کردن امکان انتقال و جستجوی المانها می‌باشد [۲۶].

هیچ یک از این استانداردها مربوط به فشرده‌سازی نمی‌باشند و فقط به منظور کامل کردن بحث استانداردها در اینجا ذکر شده‌اند.

فصل سوم

مبانی تبدیل موجک

۳-۱ چرا تبدیل موجک؟

قبل از توصیف تبدیل موجک و بیان کاربرد آن در فشرده‌سازی تصویر لازم است به دو سوال زیر پاسخ داده شود:

۱- مشکل تبدیل DCT چیست و چرا ما باید از تبدیل موجک استفاده کنیم؟

۲- اگر تبدیل موجک بهتر از DCT است چرا استاندارد JPEG از آن استفاده نکرد؟

پاسخ به اولین سوال به صورت زیر است: تبدیل DCT و دیگر تبدیلاتی که مبتنی بر بلوک عمل می‌کنند، به این شکل عمل می‌کنند که تصویر را به تعدادی بلوک ناهمپوشان تقسیم کرده و هر بلوک را جداگانه پردازش می‌کنند. در نرخ بیت‌های بسیار کوچک لازم است که ضرایب تبدیل با گام بزرگی کوانتیزه شوند و بنابراین در حین کدگشایی، خطای بازسازی بزرگی تولید می‌شود. این خطا در مرز بلوکها واضحتر و مشهودتر از بقیه نواحی بوده و تولید یک اثر نامطلوب به نام اثر بلوکی^{۷۵} می‌کند. یک راه برای حذف یا کاهش این اثر نامطلوب این است که به توابع پایه اجازه دهیم در این نقاط به صفر میل کنند یا اینکه بلوکها را به صورت همپوشان انتخاب کنیم. به تکنیک دوم، تبدیل متعامد همپوشان (${}^{\vee}LOT$) گفته می‌شود [۳]. تبدیل موجک حالت خاصی از این تبدیل است و بنابراین انتظار می‌رود اثرات بلوکی از خود نشان ندهد.

پاسخ به سوال دوم مربوط به روشهای جدیدی موجود در زمینه کدگذاری تصویر در اواسط دهه ۱۹۸۰ می‌شود یعنی زمانی که استاندارد JPEG در حال بررسی و توسعه بود. گرچه در این زمان، تبدیل موجک و نیاکان آن، کدگذاری زیرباندی، شناخته شده بودند، هنوز روش کارامدی برای کدگذاری ضرایب تبدیل موجک در مقایسه با

⁷⁵Blocking Artefact

⁷⁶Lapped Orthogonal Transform

تبدیل DCT وجود نداشت. در عمل تمام پیشنهادهای ارسال شده برای کمیته JPEG مبتنی بر DCT بوده و هیچ یک استفاده‌ای از تبدیل موجک نکرده بود. به علاوه در همان زمان از بین ۱۵ پیشنهاد ارسال شده برای کمیته H.261، ۱۴ پیشنهاد مبتنی بر DCT و ۱ پیشنهاد مبتنی بر چندی سازی برداری^{۷۷} بوده و هیچ یک اشاره‌ای به تبدیل موجک نکرده بودند. به همین دلیل بنا بر وضعیت زمانی آن سالها، از تبدیل DCT در استاندارد JPEG استفاده شد.

با این حال، وضعیت روشهای فشرده‌سازی تصویر مبتنی بر تبدیل موجک از زمان پیشنهاد استاندارد JPEG تاکنون رشد و پیشرفت فزاینده و چشمگیری داشته است. بیشترین سهم در این زمینه را باید به آقای ژوزف شاپیرو^{۷۸} داد که روش موجک درخت صفر جاسازی شده EZW را پیشنهاد داد و پیشرفت بزرگی در زمینه کدگذاری ضرایب موجک ایجاد کرد [۲۷].

۳-۲ کدگذاری زیرباندی

قبل از پرداختن به تبدیل موجک، نگاهی به نیاکان آن یعنی کدگذاری زیرباندی می‌اندازیم که گاهاً به آن تبدیل موجک اولیه نیز گفته می‌شود [۲۸]. البته همان طور که بعداً خواهیم دید این دو اصطلاح، یکی هستند. اصطلاح زیرباند محصولی حاصل کار مهندسین [۲۹] و تبدیل موجک حاصل کار ریاضی‌دانان [۳۰] است. بنابراین، قبل از پرداختن به ریاضیات که گاهاً تعقیب و ادامه آن خسته کننده می‌شود، نگاهی مهندسی به پردازش چند درجه‌ی تفکیک^{۷۹} سیگنال می‌کنیم تا درک موضوع برای ما آسانتر گردد.

کدگذاری زیرباندی اولین بار توسط کروچر و دیگران در سال ۱۹۷۶ [۲۹] پیشنهاد گردید و تاکنون قدرت و کارایی آن در کدگذاری صوت و تصویر به عنوان یک ابزار ساده و قدرتمند به اثبات رسیده است. ایده اصلی این

⁷⁷Vector Quantization

⁷⁸Jussef Shapiro

⁷⁹Multiresolution

تکنیک، بخش‌بندی طیف سیگنال به چندین محدوده یا باند فرکانسی⁸⁰ و سپس کدگذاری و انتقال هر باند به طور مجزا می‌باشد. این کار مختص کدگذاری تصویر ارائه و استفاده گردید. در توجیه این کار سه نکته قابل تامل است. اول این که طیف تصاویر طبیعی اغلب طیفی غیرخطی است و بیشتر انرژی اینگونه تصاویر در محدوده فرکانسهای پایین متمرکز است. نکته دوم این که میزان درک انسان از نویز در هر دو باند فرکانسی بالا و پایین افت می‌کند؛ بنابراین، طراح می‌تواند سیستم فشرده‌سازی را به گونه‌ای طراحی کند که مقدار اعوجاج حاصل از فشرده‌سازی بر طبق یک معیار ادراکی تنظیم شود. بالاخره، سوم این که از آنجا که در این روش، تصاویر به صورت کلی و یکجا پردازش می‌شوند نه بصورت بلوکی، بنابراین مساله‌ای به نام اعوجاج بلوکی مشابه با آن چه در مورد روشهای مبتنی بر تبدیل بلوکی مانند DCT مطرح بود، وجود نخواهد داشت.

بنابراین گرچه ایده زیرباندی مشابه با تبدیل فوریه، مبتنی بر تحلیل فرکانسی تصویر است اما فیلتربانکهای آن خاصیت ناهمبسته سازی بهتری دارند که این خاصیت در مورد تصاویر طبیعی مناسبتر و مطلوبتر است. برای توضیح بیشتر به این نکته توجه کنید که توابع پایه فوریه در حوزه فرکانسی کاملاً دقیق اما در حوزه مکانی دارای دقت نمی‌باشند. به عبارت دیگر انرژی آنها در سرتاسر بازه مکانی گسترده شده است. این امر در صورتیکه پیکسلهای تصویر همواره و همیشه با هم همبستگی داشته باشند، مشکلی محسوب نمی‌شود اما در عمل چنین نبوده و پیکسلهای تصویر در بازه بسیار محدودی با هم همبستگی دارند. به ویژه در نقاط ناپیوستگی تصاویر مانند لبه‌ها، بازه همبستگی مذکور بسیار کوتاه است. برخلاف توابع پایه فوریه، توابع پایه‌ی زیرباندی نه تنها تمرکز (یا دقت) فرکانسی⁸¹ نسبتاً خوبی دارند، بلکه در حوزه مکانی نیز فشرده و متمرکز می‌باشند. اگر لبه‌های تصویر بیش از حد به هم نزدیک نباشند، اغلب توابع یا عناصر پایه‌ی زیرباندی با این نواحی مرز مشترک نداشته و بنابراین به طور متوسط خاصیت ناهمبسته‌سازی خوبی از خود نشان می‌دهند (توجه شود که انجام فشرده‌سازی مستلزم حذف یا کاهش تزیاید بوده و کاهش تزیاید نیز مستلزم ناهمبسته‌سازی است).

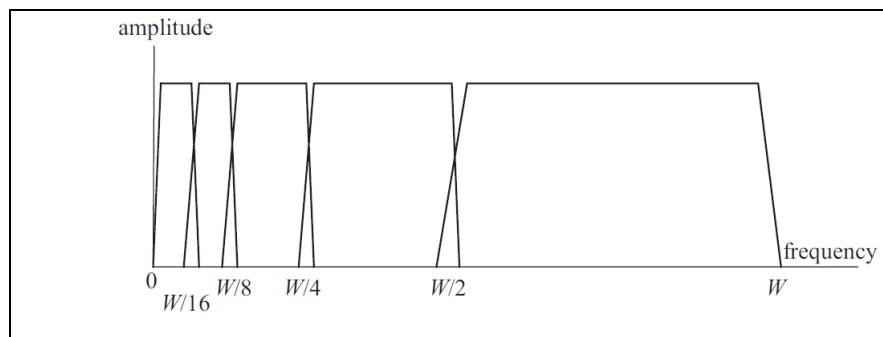
⁸⁰Frequency Band

⁸¹Frequency Concentration

در کدگذاری زیرباندی، جداسازی باندها از یکدیگر توسط فیلتر کردن تصویر ورودی به کمک بانکی از فیلترهای تحلیل میان‌گذر، مطابق با شکل ۱-۳، انجام می‌شود. در این شکل ملاحظه می‌کنید که جهت هماهنگ کردن طیف تصاویر تجزیه شده‌ی خروجی با مشخصات سیستم بینایی انسان، فیلترها به صورت باندهای اکتاو^{۸۲} (یعنی افزایش عرض باندها به صورت توان صحیحی از ۲) طراحی و تنظیم شده‌اند.

از آنجا که پهنای باند هر نسخه از تصویر فیلتر شده کاهش یافته است، از نظر تئوری می‌توان با نرخ کمتری (بر طبق قضیه نایکوئیست) نمونه برداری کرد و تعدادی زیرتصویر^{۸۳} کاهش بعد یافته به دست آورد. سپس این زیرتصویرها کوانتیزه، کدگذاری و ارسال می‌شوند. در واحد کدگشا، زیرتصویرهای دریافتی به ابعاد اولیه بازگردانده شده و از مجموعه‌ای از فیلترها به نام فیلتربانک ترکیب عبور داده می‌شوند. این فیلتربانک عمل درونیایی و ترکیب زیرتصویرها به منظور بازسازی تصویر اولیه را انجام می‌دهد.

فیلترهایی که در عمل در فیلتربانک استفاده می‌شوند، مانند شکل ۱-۳، ناحیه یا باند گذر محدود و غیرصفر دارند لذا در این فیلتربانکها در فرآیندهای زیرنمونه برداری/بالا نمونه برداری مقدار اعوجاج همپوشانی^{۸۴} در تصویر بازسازی شده وجود خواهد داشت. برای رفع مشکل اعوجاج، فیلترهای مورد استفاده باید رابطه خاصی با هم داشته باشند طوری که مولفه‌های همپوشان اثر یکدیگر را خنثی کرده و تصویری بدون اعوجاج همپوشانی حاصل شود.



⁸²Octave

⁸³Subimage

⁸⁴Aliasing Distortion

شکل ۳-۱: یک نمونه بانک فیلترهای میان گذر

۳-۳ تبدیل موجک

تبدیل موجک حالت خاصی از کدگذاری زیرباندی است که در زمینه کدگذاری تصویر و ویدیو بسیار فراگیر و متداول شده است. کدگذاری زیرباندی تصاویر مبتنی بر تحلیل فرکانسی است حال آنکه تبدیل موجک بر تئوری تقریب^{۸۵} استوار است. با این حال از آنجا که تصاویر طبیعی به طور محلی هموار بوده و می توان آن را با تقریب تکه‌ای-چندجمله‌ای مدل کرد، اگر از تابع چندجمله‌ای مناسب استفاده کنیم، عمل تقریب متناظر با نوعی تحلیل فرکانسی خواهد شد که مشابه با ایده زیرباندی است. در حقیقت، تحلیل موجک راه و ابزار بسیار موثری برای تقریب چنین توابعی در اختیار ما قرار می‌دهد طوری که از تعداد بسیار کمی عناصر پایه (برای انجام تقریب) استفاده می‌کند.

از نظر ریاضی، تبدیل موجک یک تابع مجذور-انتگرال‌پذیر $x(t)$ معادل با تجزیه این تابع برحسب مجموعه‌ای از توابع پایه به صورت زیر است:

$$X_w(a, b) = \int_{-\infty}^{\infty} x(t) \Psi_{a,b}(t) dt \quad ۱-۳$$

که در آن $\Psi_{a,b}(t)$ به نام تابع پایه شناخته می‌شود. این تابع نسخه‌ای کشیده شده و جابجا شده از یک سیگنال میان گذر مانند $\Psi(t)$ است که تابع موجک مادر^{۸۶} نامیده شده و به صورت زیر تعریف می‌شود:

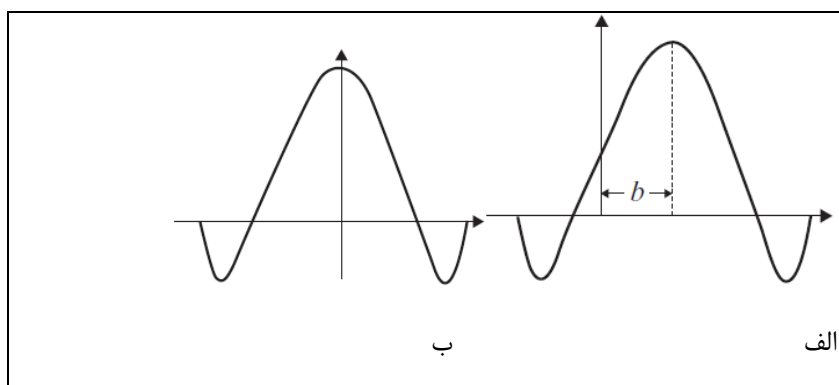
$$\Psi_{a,b} = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right) \quad ۲-۳$$

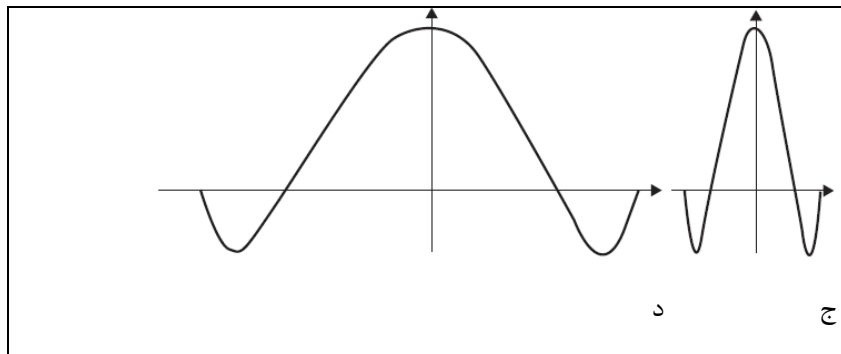
که در آن a و b به ترتیب پارامترهای اتساع^{۸۷} و جابجایی^{۸۸} نامیده می‌شوند. تاثیر این پارامترها در شکل ۳-۲ نشان داده شده است.

⁸⁵Approximation Theory

⁸⁶Mother Wavelet

عرض تابع پایه با تغییر فاکتور اتساع (یا مقیاس) a به این شکل تغییر می‌کند که هرچه این فاکتور بزرگتر شود، عرض زمانی تابع پایه نیز بزرگتر می‌شود و بنابراین، عرض (یا پهنای باند) فرکانسی آن کمتر می‌شود. بنابراین به کمک این پارامتر می‌توانیم درجه‌ی تفکیک زمانی و فرکانسی را (با مصالحه‌ای که بین یکدیگر دارند) در تبدیل موجک تغییر دهیم و همین ویژگی جالب تبدیل موجک است که باعث می‌شود این تبدیل در تحلیل سیگنالهایی که ویژگیهایی با اندازه‌های مختلف دارند، مانند تصاویر طبیعی⁸⁷، مناسب باشد. متناظر با هر اندازه‌ی ویژگی‌ای یک تابع پایه $\Psi_{a,b}(t)$ وجود دارد که آن ویژگی به بهترین وجه توسط این تابع پایه تحلیل می‌شود. برای مثال، در تصویری از یک خانه که شخصی از پنجره بیرون را نگاه می‌کند، تابع پایه‌ی متناظر با مقدار a بزرگ به راحتی کل خانه را تحلیل می‌کند اما برای تحلیل شخص و پنجره‌ی کنار آن باید از تابع مقیاسی با اندازه‌ی متوسط a استفاده کرد؛ همچنین، برای تحلیل نواحی مربوط به چشم شخص باید از تابع مقیاسی با اندازه کوچک a استفاده نمود. بنابراین، تبدیل موجک معادل با تحلیل سیگنال با فیلترهای میان‌گذری است که فرکانس میانی آنها متغیر و وابسته به پارامتر a اما فاکتور کیفیت⁸⁹ آنها ثابت است. توجه کنید که فاکتور کیفیت یک فیلتر میان‌گذر، نسبت فرکانس مرکزی به پهنای باند آن فیلتر تعریف می‌شود.

⁸⁷Dilation⁸⁸Translation⁸⁹Natural Images⁹⁰Quality Factor



شکل ۳-۲: تاثیر اتساع و جابجایی زمانی روی موجک مادر؛ (الف): موجک مادر $\Psi(t) = \Psi_{1,0}(t)$ با $a=1$ و $b=0$ ؛ (ب) حالت $a=1$ و $b \neq 0$ ؛ (ج): حالت $a=2$ و $b=0$ ؛ و (د): حالت $a=1/2$ و $b=0$.

۳-۳-۱ تبدیل موجک گسسته

تبدیل موجک تعریف شده در رابطه‌ی ۳-۱ سیگنال یک بعدی $x(t)$ را به تابعی دوبعدی مانند $X_w(a, b)$ نگاشت می‌کند و بنابراین موجب تولید تزايد زیادی می‌شود. سیگنال اولیه را می‌توان از روی تبدیل موجک و به ازاء مقادیر گسسته‌ای از پارامترهای a و b به دست آورد [۹]. پارامتر a را می‌توان با انتخاب $a = a_0^m$ (که $a_0 > 1$ و m عددی صحیح است)، گسسته کرد. با افزایش مقدار a پهنای باند (یا همان درجه تفکیک فرکانسی) تابع پایه کاهش می‌یابد و بنابراین برای پوشش یک ناحیه‌ی فرکانسی مشخص، نیاز به گامهای بیشتری (به نام سلولهای درجه تفکیک) است. به طور مشابه، گسسته‌سازی پارامتر b نیز معادل با نمونه‌برداری زمانی است که فرکانس نمونه‌برداری وابسته به پهنای باند سیگنال نمونه‌برداری شده دارد که این پهنای باند نیز به نوبه خود به طور معکوس متناسب با پارامتر a است. پارامتر b را میتوان به صورت $b = nb_0 a_0^m$ انتخاب کرد. اگر $a_0 = 2$ و $b_0 = 1$ انتخاب شوند، برخی از درجات آزادی انتخاب تابع $\Psi(t)$ وجود دارند که موجب می‌شوند توابع $\Psi_{m,n}(t)$ تشکیل یک پایه‌ی یک-متعامد^{۹۱} در فضای توابع مجذور-انتگرال‌پذیر دهند. این امر به این معنا است که هر تابع مجذور-انتگرال‌پذیر $x(t)$ را می‌توان به صورت یک ترکیب خطی از توابع پایه به صورت زیر توصیف کرد:

^{۹۱}Orthonormal

$$x(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \alpha_{m,n} \Psi_{m,n}(t) \quad 3-3$$

که در آن، ضرایب $\alpha_{m,n}$ به نام ضرایب تبدیل موجک تابع $x(t)$ نامیده شده و از روی رابطه‌ی ۱-۳ چنین محاسبه می‌شوند:

$$\alpha_{m,n} = \int_{-\infty}^{\infty} x(t) \Psi_{m,n}(t) dt \quad 4-3$$

جالب است توجه شود که با هر افزایش مقدار m مقدار a دو برابر می‌شود (یعنی دوبرابر شدن عرض یا پهنای باند زمانی و نصف شدن عرض یا پهنای باند فرکانسی). این ویژگی متناظر با تحلیل سیگنال به کمک تجزیه‌ی اکتاو باندهای فرکانسی و نیز تبدیل موجک دوتایی^{۹۲} است. با توجه به شکل ۱-۳ (که برای توضیح اصول عملکرد کدگذاری زیرباندی استفاده شد)، می‌توان نتیجه گرفت که تبدیل موجک در واقع نوعی کدگذاری زیرباندی است.

۳-۳-۲ نمایش چنددرجه تفکیک

با استفاده از مفهوم تحلیل چنددرجه تفکیک سیگنال^{۹۳} می‌توان کاربرد تبدیل موجک در کدگذاری تصویر را بهتر درک کرد. فرض کنید تابعی مانند $\Phi(t)$ وجود دارد طوری که مجموعه توابع $\Phi(t-n), n \in Z$ یک-متعامد باشند. همچنین فرض کنید که تابع $\Phi(t)$ جواب یک معادله تفاضلی دومقیاسی به صورت زیر باشد:

$$\Phi(t) = \sum_{n=-\infty}^{\infty} c_n \sqrt{2} \Phi(2t-n) \quad 5-3$$

که در آن

$$c_n = \int_{-\infty}^{\infty} \Phi(t) \sqrt{2} \Phi(2t-n) dt \quad 6-3$$

می‌باشد.

⁹²Dyadic Wavelet Transform

⁹³Multiresolution Signal Analysis

فرض کنید که $x(t)$ یک سیگنال مجذور-انتگرال پذیر باشد که بتوان آن را به صورت یک ترکیب خطی از توابع $\Phi(t - n)$ توصیف کرد:

$$x(t) = \sum_{n=-\infty}^{\infty} c_n \Phi(t - n) \quad ۷-۳$$

که در آن ضرایب بسط بوده و حاصل افکنش^{۹۴} سیگنال $x(t)$ بر روی توابع $\Phi(t - n)$ می باشند. از آنجا که اتساع^{۹۵} یک تابع، درجه تفکیک آن را تغییر می دهد، می توان سیگنال $x(t)$ را بر حسب درجات تفکیک مختلف (که با اتساع و فشردن تابع $\Phi(t)$ ایجاد می شوند) نمایش داد. بنابراین می توان سیگنال $x(t)$ را در هر درجه ی تفکیک m به صورت زیر نمایش داد:

$$x_m(t) = 2^{-m/2} \sum_n c_n^m \Phi(2^{-m}t - n) \quad ۸-۳$$

اگر فضای تشکیل شده از توابع $2^{-m/2} \Phi(2^{-m}t - n)$ را V_m بنامیم، با توجه به رابطه ی ۳-۵ می توان دید که تابع $\Phi(t)$ به گونه ای است که به ازاء هر $i > j$ ، هر تابعی که متعلق به فضای V_i می باشد متعلق به فضای V_j نیز می باشد و بدین ترتیب برای $i > j$ داریم $V_i \subset V_j$. بنابراین، فضاهای متناظر با مقیاسهای متوالی تودرتو بوده و هر فضای V_m (با m در حال افزایش) را می توان به صورت فضایی که درجه ی تفکیک آن رو به کاهش است، نگاه کرد.

در نتیجه، فضای متناظر با درجه ی تفکیک درشت تر^{۹۶} V_{j-1} را به دو زیرفضا می توان تجزیه کرد: یک فضا متناظر با درجه ی تفکیک ریزتر^{۹۷} و فضای دیگر، فضای W_j که فضای مکمل متعامد فضای V_j بوده و به گونه ای است که $V_j + W_j = V_{j-1}$ گردد. در این رابطه (بدلیل متعامد بودن دو فضای V_j و W_j) داریم $W_j \perp V_j$. فضای W_j در واقع فضای ناشی از اختلافات بین درجات تفکیک نرم و درشت مذکور است و

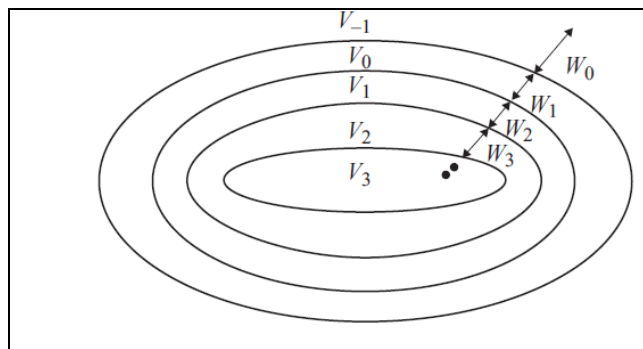
^{۹۴}Projection

^{۹۵}Dilation

^{۹۶}Coarser Resolution

^{۹۷}Finer Resolution

می‌توان از آن به صورت جزئیات لازم برای حرکت از فضای با درجه تفکیک کوچکتر، V_J ، به فضای با درجه تفکیک بزرگتر، V_{J-1} یاد کرد. سلسه مراتب فضاها از دیدگاه درجه تفکیک در شکل ۳-۳ نشان داده شده است.



شکل ۳-۳ فضاهای چند درجه‌ی تفکیک

آقای مَلْت [۱۰]⁹۸ نشان داد که در حالت کلی، پایه‌ی متناظر با فضای W_J را می‌توان به صورت جابجاییها و اتساعهای یک تابع نمونه^{۹۹} به نام موجک^{۱۰۰} توصیف کرد. بنابراین، فضای W_m در واقع فضای تشکیل شده از توابع $\Psi_{m,n}(t) = 2^{-m/2}\Psi(2^{-m}t - n)$ می‌باشد. موجک $\Psi(t) \in V_{-1}$ را می‌توان به صورت زیر از روی تابع $\Phi(t)$ به دست آورد:

$$\Psi(t) = \sum_{n=-\infty}^{\infty} (-1)^n c_{1-n} \sqrt{2} \Phi(2t - n) \quad 9-3$$

تابع $\Phi(t)$ را تابع مقیاس^{۱۰۱} مربوط به نمایش چنددرجه‌ی تفکیک می‌نامند. بنابراین ضرایب تبدیل موجک رابطه‌ی ۴-۳ متناظر با حاصل افکنش سیگنال $x(t)$ بر روی فضای جزئیات متناظر با درجه‌ی تفکیک m ، یعنی W_m می‌باشند. بنابراین، تبدیل موجک اساساً یک سیگنال را به فضاهای با درجات تفکیک مختلفی تجزیه می‌کند. در متون علمی، به این نوع تجزیه در حالت کلی، تجزیه‌ی چنددرجه‌ی تفکیک^{۱۰۲} اطلاق می‌گردد.

⁹⁸Mallat

⁹⁹Prototype

¹⁰⁰Wavelet

¹⁰¹ Scaling Function

¹⁰² Multiresolution Decomposition

در ادامه مثالی از نحوه‌ی محاسبه‌ی موجک هار^{۱۰۳} به کمک تکنیک فوق، آورده می‌شود.

مثال (موجک هار)

تابع مقیاس متناظر با موجک هار همان تابع معروف مستطیلی است:

$$\Phi(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

تابع فوق در رابطه‌ی ۵-۳ صدق کرده و ضرایب c_n مربوطه را می‌توان از رابطه‌ی ۶-۳ به صورت

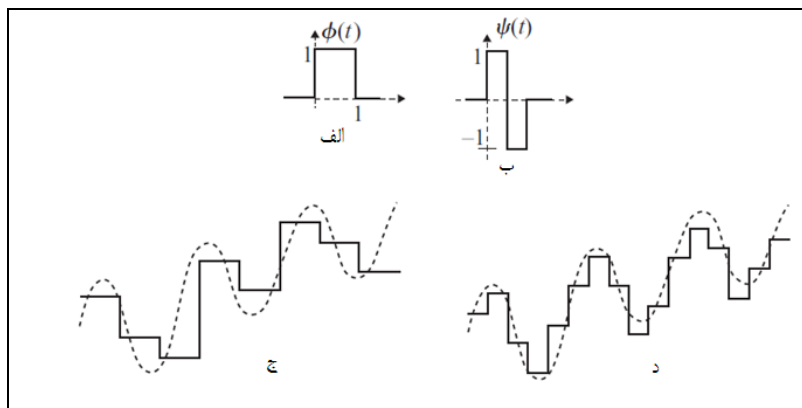
$$c_n = \begin{cases} \frac{1}{\sqrt{2}} & n = 0, 1 \\ 0 & \text{otherwise} \end{cases}$$

محاسبه کرد. بنابراین با استفاده از رابطه‌ی ۹-۳ می‌توان موجک هار را به صورت زیر به دست آورد:

$$\Psi(t) = \Phi(2t) - \Phi(2t - 1)$$

$$\Psi(t) = \begin{cases} 1 & 0 \leq t < \frac{1}{2} \\ -1 & \frac{1}{2} \leq t < 1 \end{cases}$$

تابع مقیاس $\Phi(t)$ (تابع مستطیلی) و موجک هار متناظر با آن به ترتیب در شکل‌های ۳-۴-الف و ب نشان داده شده‌اند.



شکل ۳-۴ (الف) تابع مقیاس هار، (ب) موجک هار، (ج) تقریب یک تابع پیوسته، $x(t)$ ، در مقیاس درشت تر $A_0x(t)$ و

(د) مقیاس نرمتر (یا بزرگتر) $A_1x(t)$

از دیدگاه تقریب (سیگنال) می توان تجزیه ی چنددرجه ی تفکیکی را چنین توضیح داد. فرض کنید سیگنال $x(t)$ در درجه ی تفکیک j توسط تابع $A_jx(t)$ و از طریق بسط دادن به کمک توابع پایه ی متعامد تقریب زده شود.

فضای W_j فضای اختلافات بین مقیاس درشت تر V_{j-1} و مقیاس نرمتر V_j می باشد. سیگنال $D_jx(t) \in W_j$

نیز اختلاف بین دو تقریب از سیگنال $x(t)$ در دو درجه ی تفکیک j و $j-1$ است. در واقع داریم :

$D_jx(t) = A_{j-1}x(t) - A_jx(t)$. بنابراین سیگنال $x(t)$ را می توان به دو قسمت تقسیم کرد:

$$x(t) = A_{-1}x(t) = A_0x(t) + D_0x(t)$$

شکل های ۳-۴ ج و ۳-۴ د نشان دهنده ی دو تقریب از یک تابع پیوسته در دو درجه ی تفکیک متوالی و به کمک

تابع مقیاس مستطیلی می باشند. تقریب درشت تر $A_0x(t)$ در شکل ۳-۴ ج و تقریب انجام شده در درجه ی

تفکیک بزرگتر، یعنی $A_1x(t)$ نیز در شکل ۳-۴ د نشان داده شده اند.

در یک تابع هموار و با تغییرات آهسته ی $x(t)$ عمده ی تغییرات (که همان انرژی سیگنال می باشند) در جزء

$A_0x(t)$ قرار گرفته و جزء $D_0x(t)$ تقریباً صفر می باشد. با تکرار فرآیند جداسازی فوق الذکر و انجام یک

بخش بندی دیگر به صورت $A_0x(t) = A_1x(t) + D_1x(t)$ می توان تبدیل موجک سیگنال $x(t)$ را محاسبه

کرده و تابع اولیه $x(t)$ را می توان برحسب موجک های آن به صورت زیر بسط داد:

$$x(t) = D_0x(t) + D_1x(t) + D_2x(t) + \dots + D_nx(t) + A_nx(t) \quad 10-3$$

که در آن پارامتر n بیانگر تعداد سطوح تجزیه است. از آنجا که دامنه‌ی تغییرات^{۱۰۴} سیگنالهای جزئیات $D_jx(t)$ بسیار کمتر از دامنه‌ی تغییرات تابع اولیه‌ی $x(t)$ است، کدگذاری آنها راحت‌تر از کدگذاری ضرایب بسط رابطه‌ی ۳-۳ است.

۳-۳-۳ تبدیل موجک و فیلتر بانک

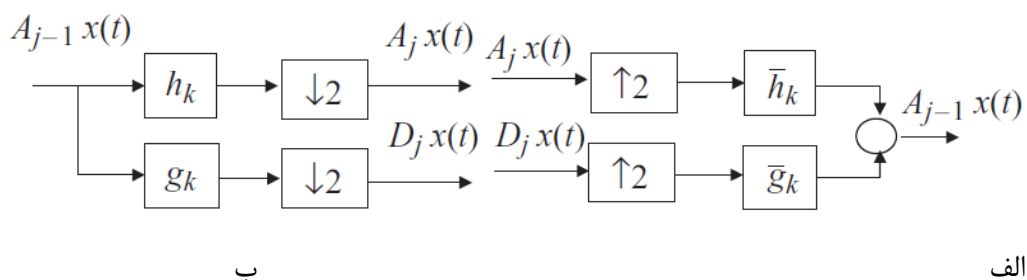
برای اینکه فرآیند جداسازی مکرر گفته شده در بخش قبل در عمل قابل پیاده‌سازی باشد باید الگوریتم موثری برای محاسبه‌ی $D_jx(t)$ از روی ضرایب بسط اولیه‌ی سیگنال $x(t)$ در دست باشد. یکی از نتایج بخش‌بندی چنددرجه‌ی تفکیک فضا این است که تابع مقیاس $\Phi(t)$ دارای خاصیت خودتشابهی است. اگر $\Phi(t)$ و $\bar{\Phi}(t)$ به ترتیب توابع مقیاس تحلیل (یعنی مربوط به مسیر تجزیه) و ترکیب (یعنی مربوط به مسیر بازسازی) باشند و همچنین توابع $\Psi(t)$ و $\bar{\Psi}(t)$ به ترتیب موجکهای تحلیل و ترکیب باشند، آن گاه از آن جا که $V_j \subset V_{j-1}$ می‌باشد، این توابع را می‌توان به صورت بازگشتی به صورت زیر تعریف کرد:

$$\begin{aligned} \Phi(t) &= \sum_{n=-\infty}^{\infty} c_n \sqrt{2} \Phi(2t - n) \\ \bar{\Phi}(t) &= \sum_{n=-\infty}^{\infty} \bar{c}_n \sqrt{2} \bar{\Phi}(2t - n) \\ \Psi(t) &= \sum_{n=-\infty}^{\infty} d_n \sqrt{2} \Phi(2t - n) \\ \bar{\Psi}(t) &= \sum_{n=-\infty}^{\infty} \bar{d}_n \sqrt{2} \bar{\Phi}(2t - n) \end{aligned} \quad 11-3$$

این روابط بازگشتی راهی را برای محاسبه ضرایب تقریب $x(t)$ در درجه تفکیک j ، یعنی $A_jx(t)$ ، و ضرایب سیگنال جزئیات، یعنی $D_jx(t)$ از روی ضرایب تقریب سیگنال در مقیاس بالاتر، یعنی $A_{j-1}x(t)$ در اختیار قرار می‌دهند. به کمک برخی روابط و دستکاریهای ریاضی می‌توان نشان داد که هر دو گروه ضرایب تقریب و

¹⁰⁴Dynamic Range

جزئیات مربوط به یک درجه تفکیک ریزتر را می‌توان از طریق کانولوشن کردن ضرایب تقریب سیگنال در درجه‌ی تفکیک درشت‌تر با یک فیلتر خاص و سپس زیرنمونه‌برداری سیگنال حاصل شده با نرخ ۲ به دست آورد. برای این کار، در مورد ضرایب تقریب مربوط به درجه تفکیک ریزتر (یا پایین‌تر^{۱۰۵}) از یک فیلتر پایین‌گذر با ضرایب $h_k = c_{-k}$ و برای جزئیات نیز از یک فیلتر بالاگذر با ضرایب $g_k = d_{-k}$ استفاده می‌شود. به عکس، می‌توان سیگنال مربوط به درجه تفکیک بالاتر (یا درشت‌تر) را می‌توان از روی سیگنال جزئیات متناظر با آن درجه‌ی تفکیک و سیگنال تقریب مربوط به درجه‌ی تفکیک پایین‌تر بازیابی کرد. برای انجام این کار ابتدا ضرایب سیگنال‌های تقریب و جزئیات گفته شده را با نرخ ۲ بالانمونه‌برداری^{۱۰۶} کرده، سپس با به ترتیب فیلترهای ترکیب^{۱۰۷} با ضرایب $\bar{h}_k = \bar{c}_k$ و $\bar{g}_k = \bar{d}_k$ کانولوشن کرده و در پایان حاصله‌های به دست آمده را با هم جمع می‌کنیم. یک مرحله از هرکدام از فرآیندهای تفکیک (یا تجزیه) و ترکیب در شکل ۳-۵ نشان داده شده است. این شکل در واقع مشابه با بلوک دیاگرام مربوط به کدگذاری زیرباندی می‌باشد. در فرآیند تجزیه، سیگنال ورودی به دو مولفه باند بالاگذر و باند پایین‌گذر تفکیک شده و در نتیجه درجه‌ی تفکیک فرکانسی با ضرایب ۲ افزایش می‌یابد گرچه به دلیل استفاده از عمل زیرنمونه‌برداری (با نرخ ۲) درجه‌ی تفکیک زمانی با ضریب ۲ کاهش می‌یابد. در نتیجه، در مجموع هر مرحله از فرآیند تجزیه موجب بهتر/بیشتر شدن درجه‌ی تفکیک فرکانسی و کمتر شدن درجه‌ی تفکیک زمانی می‌گردد.



شکل ۳-۵ یک مرحله تبدیل موجک، (الف) تحلیل (تجزیه)، (ب) ترکیب

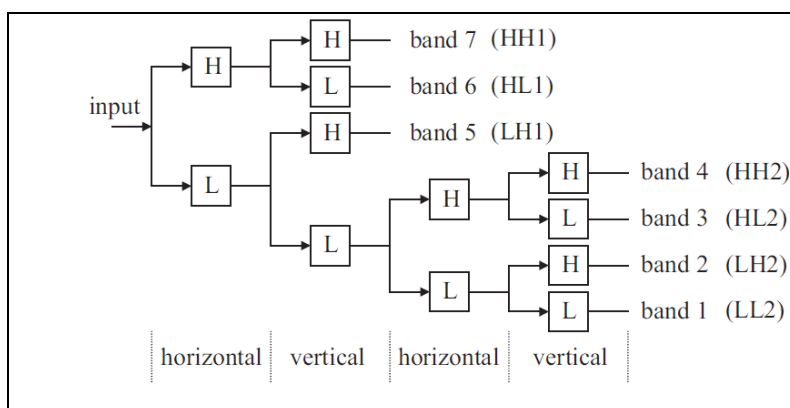
¹⁰⁵ Lower Resolution

¹⁰⁶ Up-Sampling

¹⁰⁷ Synthesis

۳-۳-۴ سیستمهای مرتبه‌ی بالاتر

با تعمیم ساختار فیلترکردن دو باندهی نشان داده شده در شکل ۳-۵ در هر بعد از ابعاد سیگنال مورد بررسی (مانند صوت و تصویر)، می‌توان به تبدیل موجک چندبعدی^{۱۰۸} دست یافت. برای مثال برای تجزیه یک تصویر (که سیگنالی دوبعدی محسوب می‌شود)، می‌توان فرآیند تجزیه‌ی یک بعدی را در هر کدام از ابعاد سطری و ستونی اعمال کرد. یک نمونه از چنین فرآیندی در شکل ۳-۶ نشان داده شده است. این شکل فرآیند مربوط به تبدیل موجک هفت-باندهی است که در آن فرآیند تجزیه‌ی باندهی یک در میان در هر یک از راستاهای افقی و عمودی به کار گرفته شده است.



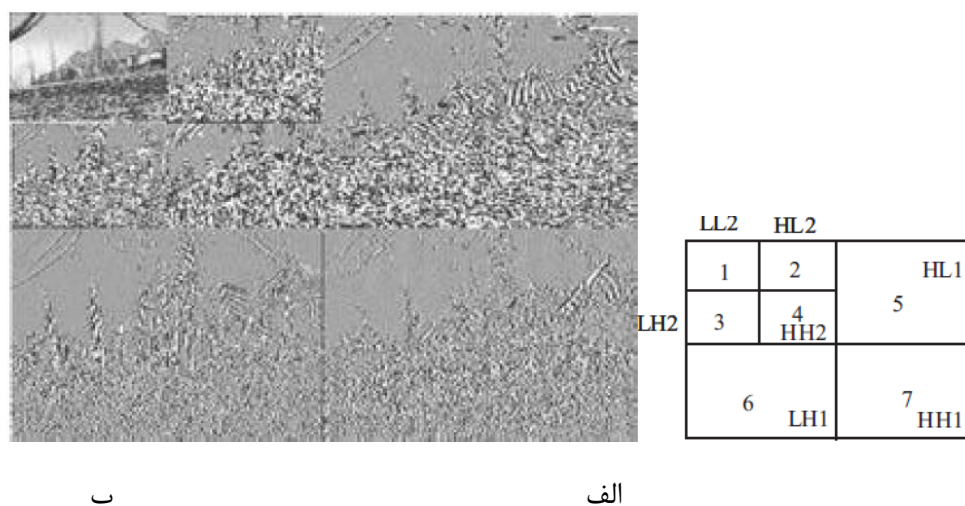
شکل ۳-۶ تبدیل موجک چندباندی به کمک تکرار تجزیه‌ی دوباندی

در شکل ۳-۶ نمادهای L و H به ترتیب بیانگر فیلترهای تجزیه‌ی پایین‌گذر و بالاگذر بوده که در آنها از زیرنمونه‌برداری با نرخ ۲ نیز استفاده می‌شود. در اولین مرحله از این فرآیند تجزیه‌ی دوتایی^{۱۰۹} سه زیرتصویر با محتویات فرکانس بالا تولید می‌شوند. برای مثال، زیرتصویر LH1 عمدتاً دارای جزئیات فرکانس بالا (در راستای عمودی و جزئیات فرکانس پایین (در راستای افقی) می‌باشد. این ویژگیها در مورد زیرتصویر HL1 برعکس می‌باشند. زیرتصویر HH1 در هر دو راستای افقی و عمودی دارای جزئیات فرکانس بالا می‌باشد. جزئیات تصویر در درجه‌ی تفکیک پایین‌تر (یعنی مرحله‌ی دوم از فرآیند تجزیه‌ی نشان داده شده در شکل ۳-۶) با نمادهای

¹⁰⁸Multidimensional Wavelet Transform

¹⁰⁹Dyadic Decomposition

LL2، HL2، LH2 و HH2 نمایش داده می‌شوند. باند LL2 یک تصویر زیرنمونه‌برداری شده‌ی پایین‌گذر است که یک نسخه‌ی رونوشت برداری (کپی) شده از روی تصویر اصلی در ابعاد کوچکتر می‌باشد. حاصل اعمال فرآیند نشان داده شده در شکل ۳-۶ روی یک نمونه تصویر در شکل ۳-۷ الف نشان داده شده است (تصویر اصلی در این شکل نشان داده نشده است). در این فرآیند، هفت زیرتصویر تولید می‌شود که نام و جایگاه هرکدام در شکل ۳-۶ ب مشخص شده است.



شکل ۳-۷ الف) هفت زیرتصویر تولید شده توسط فرآیند نشان داده شده در شکل ۳-۶، ب) چیدمان باندهای مختلف متناظر با فرآیند شکل ۳-۶.

۳-۴ کدگذاری زیرتصویرهای موجک

پایین‌ترین باند زیرتصویرهای موجک یک کپی از تصویر اولیه است که ابعاد آن، بسته به تعداد سطوح تجزیه، (مانند شکل ۳-۷ الف)، کم و بیش کاهش یافته است. اگر تعداد سطوح تجزیه زیاد باشد، عملاً همبستگی چندانی بین پیکسل‌های پایین‌ترین باند وجود ندارد. در این حالت، کدگذاری پیکسل به پیکسل، مشابه با آنچه که در استاندارد JPEG2000 مورد استفاده قرار گرفته است، به حد کافی مناسب است. از طرفی در استاندارد MPEG4 برخلاف استاندارد JPEG2000 از تعداد سطوح تجزیه چندان زیادی استفاده نشده است؛ بنابراین

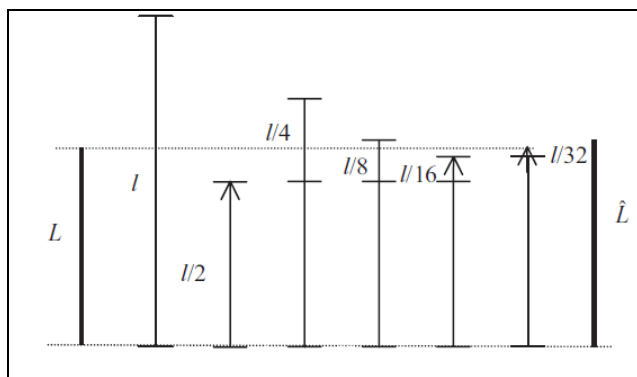
مقداری همبستگی بین پیکسل‌های باند مذکور وجود دارد. این همبستگی‌ها را می‌توان با روش کدگذاری DPCM کاهش داد.

برای کدگذاری باندهای بالاتر حاصل از تجزیه زیرباندی، ضرایب موجک مربوطه با ساختاری به نام ساختار درخت صفر¹¹⁰ (ZT) کدگذاری می‌شوند. الگوریتم [ZYEWZ] که اولین بار توسط آقای شاپیرو ارائه گردید مبتنی بر همین ساختار می‌باشد. این روش و مشتقات آن مبتنی بر دو مفهوم به نامهای کوانتیزه کردن با روش تقریب متوالی¹¹¹ و تعیین شباهت باندهای هم‌جهت می‌باشند که در ادامه توضیح داده می‌شوند.

۳-۴-۱ کوانتیزه کردن با تقریب متوالی

در این روش کوانتیزه کردن، هر ضریب تبدیل موجک بر حسب گامهای کوانتیزاسیون (به طور پیش‌رونده) کوچکتر توصیف می‌شود. تعداد مراحل تقریب بستگی به دقت مورد نظرمان در نمایش ضرایب و مقدار اعوجاج حاصله دارد.

برای دیدن این که چگونه فرآیند تقریب متوالی منجر به انجام کوانتیزاسیون می‌شود، شکل ۳-۸ را در نظر بگیرید که یک ضریب موجک به اندازه L به طور متوالی تقریب زده می‌شود تا مقدار کوانتیزه شده‌ی نهایی آن، \hat{L} ، به دست آید.



¹¹⁰Zero Tree Structure

¹¹¹Successive Approximation

شکل ۳-۸ ایده‌ی اساسی روش تقریب متوالی

فرآیند با یک مقدار طول اولیه برابر l شروع می‌شود که در حکم یک معیار برای سنجش و مقایسه است. مقدار l برابر نصف بزرگترین ضریب موجک انتخاب می‌شود. اگر ضریب موجک بزرگتر از معیار سنجش باشد، بر حسب آن نمایش داده می‌شود و الا ضریب مربوط به این مقدار از معیار سنجش برابر صفر قرار داده می‌شود. بعد از هر مرحله مقدار معیار سنجش نصف می‌شود و مقدار خطا (که برابر با اختلاف مقدار اولیه ضریب موجک و مقدار بازسازی شده‌ی آن با استفاده از ضرایبی از معیار سنجش تعریف می‌شود)، با مقدار جدید معیار سنجش مقایسه می‌شود. این فرآیند تا جایی ادامه پیدا می‌کند که مقدار خطا به حد قابل قبول برسد. با افزایش تعداد دفعات مقایسه، می‌توان مقدار خطا را به حد دلخواه کوچک کرد. برای مثال، برای شکل ۳-۸ داریم:

$$\hat{L} = 0 \times l + 1 \times \frac{l}{2} + 0 \times \frac{l}{4} + 0 \times \frac{l}{8} + 1 \times \frac{l}{16} + 1 \times \frac{l}{32} \dots = \frac{l}{2} + \frac{l}{16} + \frac{l}{32}$$

ملاحظه می‌کنید که هر ضریب به اندازه دلخواه l را می‌توان برحسب دنباله‌ای از نمادهای 0 و 1 توصیف کرد. هرچه تعداد این نمادها بیشتر شود، دقت نمایش L بیشتر و در نتیجه، میزان اعوجاج کمتر می‌شود. این روش، مشابه با نمایش باینری اعداد حقیقی است که به نام نمایش صفحه بیتی^{۱۱۲} شناخته می‌شود.

کوانتیزاسیون صفحه بیتی^{۱۱۳} شکل دیگری از تقریب متوالی است که در برخی کدهای استاندارد مانند JPEG2000 استفاده می‌شود. در این روش، ابتدا ضرایب موجک با حداکثر دقت ممکن نمایش داده می‌شوند. این دقت بستگی به درجه‌ی تفکیک پیکسلی تصویر ورودی (برای مثال ۸ بیت) و نیز بازه‌ی تغییرات^{۱۱۴} ضرایب فیلترهای موجک دارد. حال، نمادهای معرف ضرایب کوانتیزه شده، یک بیت به یک بیت در هر لحظه از زمان کدگذاری می‌شوند. در این فرآیند از پرارزش‌ترین بیت (MSB) شروع شده و به کم ارزشترین بیت (LSB) کار

¹¹² Bit Plane Representation

¹¹³ Bit Plane Quantisation

¹¹⁴ Dynamic Range

خاتمه می‌یابد. بنابراین، در حالت صفحه کوانتیزاسیون M بیتی و به فرض گام کوانتایزر Δ ، مقدار طول معیار^{۱۱۵} برابر با $2^{M-1} \Delta$ خواهد بود.

۳-۴-۲ شباهتهای بین باندها

مثالی که در شکل ۳-۷ نمایش داده شد مربوط به تبدیل موجک دومرحله‌ای (هفت باند) یک تصویر معروف به تصویر باغ گل^{۱۱۶} است. ملاحظه می‌کنید که باندهای عمودی (و البته باندهای دیگر افقی و قطری) مقیاسی از یکدیگر و شبیه به هم هستند. نکته مهم در این تصاویر این است که ضرایب کوچک و کم‌ارزش تبدیل موجک تمایل دارند که در مکانهای مشابه و متناظر هم از باندهای هم‌نوع قرار بگیرند. همچنین، لبه‌ها هم تمایل دارند که در موقعیت مشابه و متناظری نسبت به هم قرار بگیرند. با توجه به اینکه زیرتصویرهای باندهای پایین‌تر (یعنی سطوح بالاتر تجزیه) دارای ابعادی برابر نصف ابعاد باندهای بالاتر هستند، می‌توان یک نمایش درخت تریبی^{۱۱۷} از باندهای هم‌راستا^{۱۱۸} ارائه کرد. نمونه‌ای از این کار در شکل ۳-۹ برای یک تبدیل موجک سه سطحی (با ده باند) نمایش داده شده است. در این شکل هر ضریب از پایین‌ترین باند عمودی، LH3، متناظر با چهار ضرایب از باند عمودی بالاتر خود، یعنی LH2، می‌باشد که آن هم متناظر با ۱۶ ضریب از باند بالاتر خود، یعنی LH1، می‌باشد. بنابراین اگر یک ضریب در باند LH3 صفر باشد احتمال زیادی وجود دارد که فرزندان آن در باندهای بالاتر LH2 و LH1 نیز صفر باشند. این مطلب برای دو راستای دیگر (یعنی افقی و قطری) نیز صادق و برقرار است. به چنین درختی، درخت صفرها یا ZT ^{۱۱۹} گفته می‌شود. روش درخت صفرها روشی موثر برای نمایش یک گروه بزرگ صفر از ضرایب تبدیل موجک می‌باشد. کافی است ریشه‌ی درخت ZT شناسایی شود تا تمام فرزندان آن در باندهای بالاتر نادیده گرفته شوند.

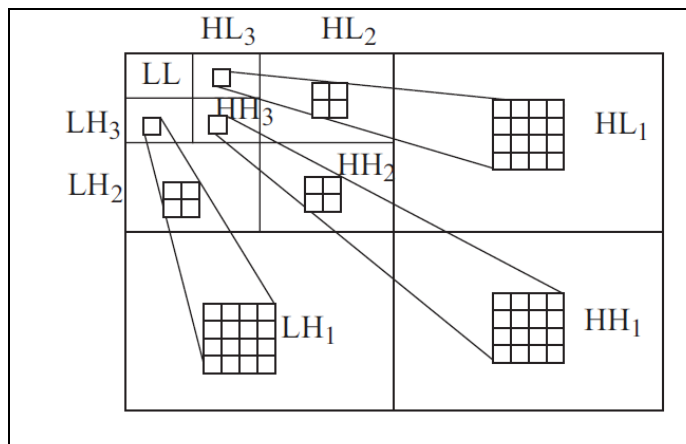
¹¹⁵ Yardstick

¹¹⁶ Flower Garden

¹¹⁷ Quad Tree

¹¹⁸ Same Orientation

¹¹⁹ Zero tree



شکل ۳-۹ نمایش درخت تربیعی از باندهای متعلق به یک راستا

فصل چهارم

در این فصل در مورد روشهای EZW، SPIHT، و EBCOT که همگی به منظور کدگذاری ضرایب تبدیل موجک استفاده می‌شوند، مطالبی آورده شده است.

۴-۱ الگوریتم EZW

با ترکیب ریشه‌های ZT و تقریب متوالی می‌توان یک ابزار کدگذاری جالب به دست آورد که نه تنها قادر به فشرده‌سازی ضرایب موجک بلکه قادر به تامین دو خواسته مقیاس پذیری مکانی^{۱۲۰} و مقیاس پذیری SNR^{۱۲۱} است [۲۷].

مراحل کدگذاری EZW با اندکی تصحیحات به صورت زیر است [۲۷]:

۱. متوسط تصویر محاسبه و از تصویر کسر می‌شود. البته این کار را می‌توان نادیده گرفت. اگر بخواهیم که پایینترین زیرباند LL را مستقل از بقیه زیرباندها کدگذاری کنیم، نیازی به انجام این کار نمی‌باشد.
۲. تبدیل موجک R مرحله‌ای (که $3R+1$ زیرباند تولید می‌کند) به تصویر (که اکنون دارای میانگین صفر شده است) اعمال می‌شود.
۳. طول اولیه میله مقایسه، l ، برابر با نصف بزرگترین ضریب موجک (از نظر قدرمطلق) قرار داده می‌شود.

¹²⁰ Spatial Scalability

¹²¹ SNR Scalability

۴. یک لیست به نام لیست غالب^{۱۲۲} تهیه می‌شود که شامل مختصات ضرایب تبدیل موجک است. این لیست بیانگر ترتیب جاروب و بررسی ضرایب موجک است. این لیست باید به گونه‌ای تنظیم شود که ضرایب متعلق به زیرباندهای فرکانس-پایینتر (یعنی مقیاس بالاتر) همواره قبل از ضرایب زیرباندهای فرکانس بالاتر جاروب و بررسی شوند. به علاوه دو لیست تهی دیگر از مختصات ضرایب به نامهای لیست زیرمجموعه^{۱۲۳} و لیست موقتی^{۱۲۴} نیز تهیه می‌شوند.
۵. تبدیل موجک تصویر جاروب و بررسی می‌شود. اگر ضریبی کوچکتر از طول فعلی میله مقایسه، l باشد به مقدار صفر بازسازی می‌گردد اما اگر بزرگتر باشد بسته به علامت آن به مقدار $\pm 3l/2$ بازسازی می‌شود.
۶. مرحله اصلی^{۱۲۵}: ضرایب بازسازی شده مجدداً با توجه به ترتیب آنها در لیست اصلی بررسی می‌شوند تا رشته‌ای از نمادها به صورت زیر تولید شود. اگر ضریب بازسازی شده مثبت یا منفی باشد، یک نماد + یا - به رشته مزبور اضافه شده و مختصات این ضریب نیز به لیست زیرمجموعه اضافه می‌شود. اگر ضریب بازسازی شده صفر باشد، مختصات آن به لیست موقتی اضافه می‌شود. در این حالت دو نماد مختلف به رشته قابل الحاق و افزودن است: اگر تمام ضرایب متناظر با ضریب فعلی در زیرباندهای هم جهت^{۱۲۶} و فرکانسهای بالاتر صفر هستند یک ریشه‌ی ZT به رشته اضافه شده و تمام ضرایب متناظر مزبور از لیست اصلی حذف و به لیست موقتی اضافه می‌شوند (چرا که اکنون مشخص شده که آنها همگی صفر می‌باشند و دیگر نیازی به جاروب و بررسی آنها نمی‌باشد) در غیر اینصورت یک صفر منفرد^{۱۲۷}، Z ، به رشته اضافه می‌شود. حال رشته‌های تولید شده توسط چهار نماد +، -، ZT ، و Z به کمک یک کدگذاری حسابی وفقی [۱۵] که مدل آن برای چهار نماد تنظیم شده باشد، کدگذاری می‌شوند. از طرفی می

122 Dominant List

123 Subordinate List

124 Temporary List

125 Dominant Pass

126 Bands of the Same Orientation

127 Isolated Zero

دانیم که در حین جاروب بالاترین باندهای فرکانسی افقی، عمودی، و قطری (یعنی HL_1 ، LH_1 و HH_1 از شکل ۳-۹) هیچ ریشه ZT قابل حصول نمی‌باشد؛ بنابراین، قبل از کدگذاری اولین ضریب این باندها، مدل کدگذار حسابی برای تنها سه ضریب $+$ ، $-$ و Z تنظیم می‌شود.

۷. طول میله مقایسه نصف می‌شود.

۸. مرحله زیرمجموعه^{۱۲۸}: ضرایبی که تاکنون به مقدار صفر بازسازی نشده‌اند، مجدداً بر طبق ترتیبی که در لیست زیرمجموعه آمده است، جاروب شده و به هر کدام مقدار $+1/2$ یا $-1/2$ طوری اضافه می‌شود که اندازه‌ی خطای بازسازی مینیمم شود. اگر $+1/2$ اضافه شده است نماد $+$ و در غیر اینصورت نماد $-$ به رشته اضافه می‌شود. در پایان مرحله زیرمجموعه، لیست زیرمجموعه به گونه‌ای دوباره‌چینی می‌شود که ضرایبی که قدرمطلق مقدار بازسازی شده‌شان بزرگتر است زودتر ظاهر شوند. حال از یک کدگذار حسابی که برای دو نماد $+$ و $-$ تنظیم شده است برای کدگذاری این دنباله از نمادها استفاده می‌شود.

۹. لیست زیرمجموعه با لیست موقتی جایگذاری شده و پس از آن لیست موقتی، خالی می‌شود.

۱۰. حال تمام فرآیند ذکر شده تاکنون از مرحله ۵ به بعد تکرار می‌شود. این فرآیند تا جایی تکرار می‌شود که تعداد بیت‌های خروجی تولید شده بیشتر از نرخ بیت مجاز شود.

در مرحله اصلی (مرحله ۶) باید به نکته‌ای توجه کرد. در این مرحله تنها مقادیر بازسازی شده‌ی ضرایبی تحت تاثیر قرار می‌گیرند که هنوز همچنان در لیست اصلی قرار داشته باشند. بنابراین برای افزایش تعداد ریشه‌های ZT ، می‌توان ضرایبی که متعلق به لیست اصلی نمی‌باشند را صفر در نظر گرفت تا تعیین کرد که یک ضریب صفر آیا (از نوع) یک ریشه ZT است یا اینکه یک صفر منفرد می‌باشد.

دنباله بیتی خروجی شامل سرآیندی^{۱۲۹} است که شامل اطلاعات مکمل و لازم برای واحد کدگشا می‌باشد. از جمله تعداد سطوح تجزیه موجک، ابعاد تصویر اصلی، مقدار اولیه میله مقایسه، و مقدار میانگین تصویر. هر دو

¹²⁸Subordinate Pass

¹²⁹Header

واحد کدگذار و کدگشا در شروع کار خود، لیست اصلی یکسانی دارند. واحد کدگشا، به موازات کدگشایی دنباله بیتی، تصویر بازسازی شده و همچنین لیستهای موقتی و زیرمجموعه خود را به روز رسانی می‌کند. به این ترتیب، واحد کدگشا قادر است مراحل را که واحد کدگذار طی کرده است دقیقاً دنبال و دنباله بیتی دریافتی را به طور صحیح کدگشایی کند. لازم است به این نکته مهم توجه شود که ترتیب مربوط به لیست زیرمجموعه در مرحله ۸ تنها مبتنی بر مقادیر بازسازی شده ضرایب می‌باشد که این مقادیر بازسازی شده در اختیار واحد کدگشا نیز می‌باشد. اگر چنین نمی‌بود واحد کدگشا قادر به دنبال کردن فعالیتهای واحد کدگذار و در نتیجه کدگشایی صحیح دنباله بیتی دریافتی نمی‌بود.

۴-۱-۱ تحلیل الگوریتم EZW

برخی نکات جالب توجه روش EZW از این قرار است:

۱. استفاده از ZTها موجب استخراج شباهت بین باندهای هم جهت و نیز کاهش تعداد نمادهای لازم جهت کدگذاری می‌شود.
۲. استفاده از یک الفبای بسیار کوچک برای نمایش یک تصویر (ماکزیمم تعداد ۴ نماد) موجب افزایش قابل توجه کارایی کدگذار حسابی می‌شود زیرا موجب می‌شود این کدگذار خود را بسیار بهتر با هرگونه تغییرات نمادها هماهنگ کند.
۳. از آنجا که بیشترین حد اعوجاج هر ضریب در هر مرحله محدود به طول فعلی میله مقایسه است، بنابراین متوسط سطح اعوجاج در هر مرحله نیز توسط طول فعلی میله مقایسه تعیین می‌شود که برای تمام باندها یکسان است.
۴. در هر مرحله تنها ضرایب بزرگتر از طول فعلی میله مقایسه به عنوان ضرایب غیرصفر کدگذاری می‌شوند. بنابراین، تمایل روش EZW به کدگذاری ضرایب بزرگتر قبل از ضرایب کوچکتر دارد. به

عبارت دیگر، روش EZW در فرآیند کدگذاری خود، اولویت را به مهمترین اطلاعات می‌دهد. این کار به کمک مرتب کردن لیست زیرمجموعه در مرحله شماره ۸ انجام می‌شود. در نتیجه، با معلوم و محدود بودن نرخ بیت، بیتها در جاهایی که بیشترین نیاز وجود دارد مصرف می‌شوند.

۵. از آنجا که روش EZW از فرآیند تقریب متوالی تبعیت می‌کند، اضافه شدن یک نماد جدید (شامل +، -، Z، و ZT) در هر زمان به رشته، موجب دقیق تر شدن و کاملتر شدن تصویر بازسازی شده می‌شود. به علاوه از آنجاییکه با ورود هر نماد بلافاصله کدگذاری و به دنباله بیتی خروجی افزوده می‌شود، بنابراین واحدهای کدگذار و کدگشا را می‌توان در هر مرحله متوقف ساخت و تصویر متناظر با اطلاعات تا آن لحظه از زمان را بازسازی کرد. به عبارت دیگر فرآیند کدگذاری و کدگشایی را می‌توان در صورتی که دنباله بیتی خروجی از نرخ بیت موجود سرریز کند، متوقف ساخت و بنابراین کنترل دقیقی روی نرخ بیت خروجی داشت. نکته دیگر این که با توجه به این که همواره مهمترین اطلاعات زودتر کدگذاری و ارسال می‌شوند (مطابق با بند قبلی)، بنابراین، نقطه یا زمانی که فرآیند کدگذاری/کدگشایی متوقف می‌شوند مهم نیست و همواره می‌توان مطمئن بود که تصویری با بهترین کیفیت ممکن تا آن لحظه از زمان در دست خواهد بود.

۶. **مقیاس پذیری مکانی و مقیاس پذیری SNR** : برای دستیابی به مقیاس پذیری مکانی و مقیاس پذیری SNR دو نوع مختلف جاروب تصویر پیشنهاد شده است. برای دستیابی به مقیاس پذیری مکانی، ضرایب تبدیل موجک به صورت زیرباند-به-زیرباند و از پایین ترین زیرباند فرکانسی به بالاترین زیرباند فرکانسی جاروب می‌شوند. برای دسترسی به مقیاس پذیری SNR ضرایب موجک در هر شاخه از درخت از بالا به پایین جاروب می‌شوند. نوع روش به کار گرفته برای جاروب کردن در دنباله‌ی بیتی مشخص می‌شود تا واحد کدگشا مطلع شود.

۴-۲ روش بخش‌بندی مجموعه به صورت درختهای سلسله مراتبی (SPIHT^{۱۳۰})

کارایی فشرده سازی روش EZW تا حدی به دلیل استفاده از کدگذاری حسابی است. آقایان Said و Pearlman نوعی از کدگذاری ضرایب موجک را پیشنهاد دادند [۳۱] که مبتنی بر تقریب متوالی بوده و کارایی فشرده‌سازی آن حتی بدون استفاده از کدگذاری حسابی بیشتر از روش EZW است. آنها نام روش خود را بخش‌بندی مجموعه در قالب درختهای سلسله مراتبی یا SPIHT انتخاب کردند. هر دو روش EZW و SPIHT روشهای کدگذاری مکانی مبتنی بر درخت هستند که از همبستگی بین ضرایب موجود در باندهای مختلف بهره می‌برند. هر کدام از این دو روش دنباله‌ی بیتی‌ای تولید می‌کنند که توسط کدگذاری صفحات بیتی^{۱۳۱} مربوط به زیرباندهای کوانتیزه شده سطوح مختلف تجزیه موجک حاصل شده‌اند. هر دوی این روشها از بررسی اهمیت مجموعه ضرایب به منظور جداسازی موثر ضرایب و در نتیجه کدگذاری ضرایب با اندازه بزرگتر استفاده می‌کنند. با این حال بخشهای مهم، حیاتی و متمایز کننده روش SPIHT شامل چگونگی بخش بندی ضرایب موجک و نوع اطلاعات مهمی است که این ضرایب با خود حمل می‌کنند.

یکی از ویژگیهای اصلی این روش این است که مسیر اجرایی الگوریتم توسط نتایج مقایسه‌های انجام شده در نقاط انشعاب آن تعیین می‌شود. به عبارت دیگر اگر هر دو واحد کدگذار و کدگشا از یک الگوریتم مرتب سازی مشترک استفاده کنند، واحد کدگشا در صورتی که نتایج مقایسه اندازه‌ها (که در واحد کدگذار انجام می‌شود) را در اختیار داشته باشد، می‌توان همان مسیر اجرایی واحد کدگذار را تکرار کند. اطلاعات مربوط به ترتیب مسیرها^{۱۳۲} را می‌توان از مسیر اجرایی طی شده بازیابی کرد.

الگوریتم مرتب سازی مجموعه ضرایب موجک، $\{C_{i,j}\}$ ، را به زیرمجموعه‌های جداسازی مانند T_m تقسیم کرده و آزمایش اندازه زیر را انجام می‌دهد:

¹³⁰ Set Partitioning in Hierarchical Trees (SPIHT)

¹³¹ Bit Planes

¹³² Ordering Information

$$\max_{(i,j) \in T_m} \{|C_{i,j}|\} \geq 2^n? \quad 1-4$$

اگر واحد کدگشا پاسخ خیر به این سوال پیدا کند (زیرمجموعه غیرمهم بوده و بنابراین) متوجه می‌شود که تمام ضرایب موجود در T_m غیرمهم می‌باشند. اما اگر پاسخ سوال فوق مثبت باشد (زیرمجموعه مهم بوده و بنابراین) از قانون مشترکی بین واحدهای کدگذار و کدگشا برای بخش بندی T_m به زیرمجموعه(های) جدید $T_{m,1}$ استفاده می‌شود و سپس آزمون اهمیت^{۱۳۳} روی تمام زیرمجموعه‌های مهم تک-مختصات^{۱۳۴} انجام شده تا ضرایب مهم شناسایی شوند.

برای کاهش تعداد مقایسه‌های اندازه ضرایب (و در نتیجه، کاهش تعداد بیت‌های پیام^{۱۳۵}) از یک قاعده بخش بندی مجموعه^{۱۳۶} استفاده می‌شود. این قاعده فرض می‌کند که ترتیب خاصی در سلسله مراتب حاصل از تجزیه زیربندی موجک وجود دارد (مشابه با شکل ۳-۹ که در کدگذاری ZT استفاده شد). در بخش ۳-۴-۲ دیدیم که چگونه می‌توان از شباهتهای بین زیرتصویرهای هم جهت در طرح هرمی تجزیه موجک برای تشکیل یک درخت جهت دار فضایی (^{137}SOT) استفاده کرد. در اینجا، هدف ایجاد بخش‌های جدید به گونه‌ای است که زیرمجموعه‌هایی که انتظار می‌رود غیرمهم باشند، شامل تعداد بسیار زیادی ضریب باشند و زیرمجموعه‌هایی که انتظار می‌رود مهم باشند، شامل تنها یک ضریب باشند.

برای روشن شدن رابطه‌ی بین مقایسه‌ی اندازه‌ها و بیت‌های پیام، از تابع زیر برای نمایش اهمیت مجموعه‌ای از مختصات T استفاده می‌شود:

¹³³Significant Test

¹³⁴Single-Coordinate

¹³⁵Message Bits

¹³⁶Set Partitioning Rule

¹³⁷Spatial Oriented Tree

$$S_n(T) = 1, \quad \text{if } \max_{\substack{(i,j) \\ (i,j) \in T}} \{|C_{i,j}|\} \geq 2^n$$

$$= 0, \quad \text{otherwise} \quad ۲-۴$$

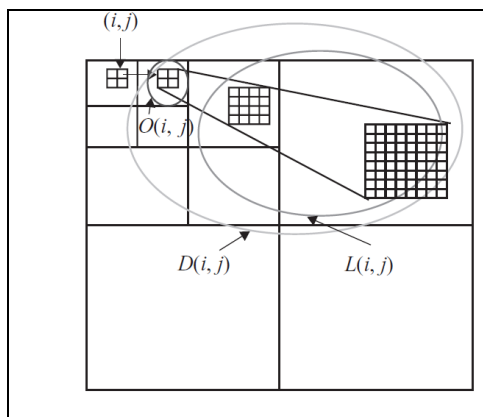
برای ساده کردن نمایش مجموعه‌های تک-مختصات، به جای $S_n(\{(i,j)\})$ از $S_n(i,j)$ استفاده می‌شود. برای دیدن این که SPIHT چگونه پیاده‌سازی می‌شود فرض کنید که $O(i,j)$ بیانگر مجموعه‌ی شامل مختصات نقاط زیرمجموعه مستقیم (یا فرزندان^{۱۳۸}) گره (i,j) باشد. برای مثال، به جز برای بالاترین و پایین‌ترین سطح سطوح هرم، می‌توان $O(i,j)$ را به صورت زیر بر حسب نسل خود توصیف کرد:

$$O(i,j) = \{(2i, 2j), (2i, 2j + 1), (2i + 1, 2j), (2i + 1, 2j + 1)\} \quad ۳-۴$$

علاوه بر این، نماد $D(i,j)$ را به عنوان مجموعه مختصات تمام نقاط زیرمجموعه مستقیم و غیرمستقیم (یا نسل^{۱۳۹}) نقطه‌ی (i,j) تعریف می‌کنیم. نماد H را نیز به صورت مجموعه شامل تمام ریشه‌های SOT (یعنی گره‌های موجود در بالاترین سطح هرم) در نظر می‌گیریم. در نهایت $L(i,j)$ را نیز به صورت زیر تعریف می‌کنیم:

$$L(i,j) = D(i,j) - O(i,j) \quad ۴-۴$$

نمایشی از $O(i,j)$ ، $D(i,j)$ و $L(i,j)$ در یک SOT در شکل ۴-۱ نشان داده شده است.



شکل ۴-۱: نمایشی از یک SOT و نحوه‌ی بخش بندی در روش SPIHT

¹³⁸Offspring

¹³⁹Descendants

با استفاده از بخشهای SOT به عنوان زیرمجموعه‌های بخش‌بندی کننده مورد استفاده در الگوریتم مرتب سازی، قوانین بخش‌بندی مجموعه را می‌توان به صورت زیر تعریف کرد:

- ✓ بخش‌بندی اولیه به کمک تمام نقاط $(i,j) \in H$ و $D(i,j)$ ‌های متناظرشان تشکیل می‌شود.
- ✓ اگر $D(i,j)$ مهم باشد آنگاه به $L(i,j)$ و چهار مجموعه‌ی تک-نقطه متناظر به صورت $(k,l) \in O(i,j)$ بخش بندی می‌شود.
- ✓ اگر $L(i,j)$ مهم باشد آنگاه به چهار مجموعه‌ی $D(k,l)$ که $(k,l) \in O(i,j)$ است، بخش بندی می‌شود.
- ✓ حال هرکدام از چهار مجموعه حاصل شده در این مرحله، به شکل و قالب مجموعه اولیه هستند و بنابراین از همان قوانین بخش‌بندی مجدداً به صورت بازگشتی استفاده می‌شود.

۴-۲-۱ الگوریتم کدگذاری

از آنجا که ترتیبی که طبق آن زیرمجموعه‌ها بررسی و آزمایش می‌شوند مهم است، به منظور امکان پیاده‌سازی عملی الگوریتم، اطلاعات مربوط به اهمیت در سه لیست مرتب شده به نامهای لیست مجموعه‌های غیرمهم (یا LIS^{۱۴۰})، لیست پیکسل‌های غیرمهم (یا LIP^{۱۴۱})، و لیست پیکسل‌های مهم (یا LSP^{۱۴۲}) ذخیره می‌شوند. در تمام این لیست‌ها هر عضو آنها به صورت مختصاتی مانند (i,j) نمایش داده می‌شود. این مختصات در لیست‌های LIP و LSP معرف پیکسها به طور انفرادی، و در لیست LIS معرف مجموعه‌ی $D(i,j)$ و یا $L(i,j)$ می‌باشد که برای ایجاد تمایز و سهولت شناساندن بین این دو نوع مجموعه، گفته می‌شود که هر عضو مجموعه‌ی LIS یا از نوع A است و یا از نوع B که به ترتیب معرف مجموعه‌های $D(i,j)$ و $L(i,j)$ می‌باشد [۳۱].

¹⁴⁰List of Insignificant Sets

¹⁴¹List of Insignificant Pixels

¹⁴²List of Significant Pixels

در طی مرحله‌ی مرتب سازی (یا چیدن)^{۱۴۳} پیکسل‌هایی از LIP که در مرحله‌ی قبل غیر مهم تشخیص داده شده بودند، بررسی شده و آنهایی که مهم (تشخیص داده) می‌شوند به لیست LSP منتقل می‌شوند. به طور مشابه، مجموعه‌ها بر طبق ترتیبی که در LIS آمده ارزیابی شده و زمانی که یک مجموعه مهم تشخیص داده می‌شود، از لیست حذف شده و بخش بندی می‌شود. حال زیرمجموعه‌های جدید حاصل شده اگر دارای بیش از یک عضو باشند به لیست LIS برگردانده می‌شوند. اما اگر این زیرمجموعه‌های جدید دارای تنها یک عضو باشند بر حسب این که غیرمهم یا مهم باشند به ترتیب به انتهای لیستهای LIP و LSP اضافه می‌شوند. لیست LSP شامل مختصات پیکسل‌هایی است که در مرحله‌ی اصلاح^{۱۴۴} قبلاً بررسی شده بودند.

بنابراین الگوریتم کدگذاری SPIHT را به صورت زیر می‌توان خلاصه نمود:

۱. مقداردهی/اولیه: مقدار اولیه طول میله مقایسه، n ، را به صورت $n = \lfloor \log_2(\max_{(i,j)} \{C_{i,j}\}) \rfloor$ انتخاب کنید.

مجموعه‌ی LSP را یک مجموعه تهی قرار دهید. مختصات $(i,j) \in H$ را به مجموعه‌ی LIP اضافه

کنید. همچنین پیکسل‌هایی که دارای نسل^{۱۴۵} می‌باشند را به عنوان نوع A به لیست LIS اضافه کنید.

۲. مرحله مرتب سازی (یا چیدن):

۱.۲. برای هر عضو (i,j) متعلق به لیست LIP کارهای زیر را انجام دهید:

۱.۱.۲. $S_n(i,j)$ را به عنوان خروجی ارسال کنید.

۲.۱.۲. اگر $S_n(i,j)=1$ است آنگاه مختصات (i,j) را به لیست LSP اضافه کرده و

علامت ضریب $C_{i,j}$ را ارسال کنید.

۲.۲. برای هر عضو (i,j) متعلق به لیست LIS کارهای زیر را انجام دهید:

۱.۲.۲. اگر عضو از نوع A است آنگاه:

۱.۱.۲.۲. $S_n(D(i,j))$ را ارسال کنید.

¹⁴³Sorting

¹⁴⁴Refinement Pass

¹⁴⁵Descendent

۲.۱.۲.۲. اگر $S_n(D(i,j))=1$ است آنگاه:

۱.۲.۱.۲.۲. برای هر $(k,l) \in O(i,j)$ کارهای زیر را انجام دهید:

۱.۱.۲.۱.۲.۲. $S_n(k,l)$ را ارسال کنید.

۲.۱.۲.۱.۲.۲. اگر $S_n(k,l)=1$ است آنگاه (k,l) را به LSP اضافه

کرده و علامت $C_{k,l}$ را ارسال کنید.

۳.۱.۲.۱.۲.۲. اگر $S_n(k,l)=0$ است آنگاه (k,l) را به انتهای LIP

اضافه کنید.

۲.۲.۱.۲.۲. اگر $L(i,j) \neq \Phi$ است آنگاه (i,j) را به انتهای LIS و به عنوان

نوع B اضافه کنید. در غیر اینصورت عضو (i,j) را از LIS حذف نمایید.

۲.۲.۲. اگر عضو از نوع B است آنگاه :

۱.۲.۲.۲. $S_n(L(i,j))$ را ارسال کنید.

۲.۲.۲.۲. اگر $S_n(L(i,j))=1$ است آنگاه :

۱.۲.۲.۲.۲. هر $(k,l) \in O(i,j)$ را به انتهای LIS به عنوان نوع A اضافه کنید.

۲.۲.۲.۲.۲. (i,j) را از LIS حذف نمایید.

۳. مرحله اصلاح: برای هر عضو (i,j) متعلق به LSP، به جز آنهایی که مورد استفاده آخرین مرحله مرتب

سازی (یعنی با شماره n مشترک) بودند، بیت MSB n ام از مقدار قدر مطلق $|C_{i,j}|$ را ارسال کنید.

۴. به روز رسانی گام کوانتیزاسیون: مقدار n را یک واحد کاهش داده و به مرحله ۲ بروید.

یک ویژگی مهم الگوریتم فوق این است که عناصری که در فوق به انتهای LIS افزوده می‌شوند، قبل از اینکه

مرحله مرتب سازی همان نوبت تکرار به پایان برسد، ارزیابی و بررسی می‌شوند. بنابراین وقتی گفته می‌شود که

«برای هر عنصری متعلق به لیست LIS» منظور آن عناصری است که به انتهای این لیست افزوده شده‌اند.

همچنین، مشابه با EZW، نرخ بیت را دقیقاً می‌توان کنترل کرد چرا که اطلاعات ارسالی از بیت‌های منفرد تشکیل شده است. واحد کدگذار می‌تواند میزان کاهش اعوجاج را به موازات ارسال اطلاعات تخمین زده و کار را در یک مقدار مورد نظر از اعوجاج پایان دهد.

در الگوریتم SPIHT به این نکته توجه کنید که واحد کدگذار بر حسب وضعیت و مقادیر ضرایب موجک، در حقیقت، تمام نقاط انشعاب (یعنی نتیجه‌ی مقایسه‌های انجام شده) را به عنوان خروجی ارسال می‌کند. بنابراین برای این که الگوریتم کاری واحد کدگشا (که همان مسیر اجرایی واحد کدگذار را دنبال و رعایت می‌کند)، به دست آوریم کافی است دنباله‌ی خروجی واحد کدگذار را به عنوان دنباله‌ی ورودی فرض کنیم. در واحد کدگشا، زمانی که مختصات ضرایب مهم به انتهای لیست LSP اضافه می‌شود، می‌توان اطلاعات مربوط به ترتیب و دوباره چینی را (که در واحد کدگذار استفاده شده بود) به دست آورد. در هنگام ورود داده‌های واحد کدگذار، سه لیست کنترلی (LIP، LIS، و LSP) همان وضعیتی را دارند که لیست‌های متناظر در واحد کدگذار در زمان شروع به کار داشتند.

کار اضافه‌ای که واحد کدگشا انجام می‌دهد این است که مرتباً تصویر بازسازی شده را به روز رسانی می‌کند. برای یک مقدار مشخص n ، هنگامی که یک مختصات به لیست LSP منتقل می‌شود، مشخص می‌شود که $2^n \leq |C_{i,j}| < 2^{n+1}$ می‌باشد. بنابراین از این دانسته و نیز علامتی که بلافاصله واحد کدگذار ارسال کرده است برای تقریب و بازسازی ضریب فعلی $C_{i,j}$ به صورت $\bar{C}_{i,j} = \bar{1.5} \times 2^n$ استفاده می‌شود. به طور مشابه در حین مرحله اصلاح، واحد کدگشا پس از دریافت بیت‌های مربوط به نمایش باینری $|C_{i,j}|$ مقدار 2^n را با $\bar{C}_{i,j}$ جمع یا تفریق می‌کند. به این ترتیب، در هر مرحله اصلاح و مرتب سازی، مقدار اعوجاج به تدریج کاهش می‌یابد.

در پایان خوب است برخی تفاوت‌های بین روش‌های EZW و SPIHT بیان و مطرح گردند. اولین تفاوت این است که آنها SOT هایی استفاده می‌کنند که اندکی با یکدیگر متفاوت هستند. در EZW هر گره ریشه موجود در بالاترین باند LL سه فرزند دارد که متعلق به یک سطح تجزیه مشترک و هر یک متعلق به یکی از زیرباندهای فرکانس بالا می‌باشند. بقیه ضرایب هریک دارای چهار فرزند در سطح تجزیه پایینتر از خود و در همان جهت

فرکانسی خود می‌باشند. اما در SPIHT، در یک گروه 2×2 از گره‌های ریشه در بالاترین باند LL، گره موجود در بالا سمت چپ دارای هیچ نسلی نبوده و بقیه سه ضریب، هر یک دارای چهار فرزند در باند فرکانس بالا و در همان جهت فرکانسی خود می‌باشند. بنابراین در SPIHT از تعداد درختان کمتر اما درختانی با تعداد عناصر بیشتر نسبت به EZW استفاده می‌شود.

یک تفاوت دیگر دو روش فوق الذکر، قوانین بخش‌بندی مجموعه‌ها است که در هر یک استفاده می‌شود. روش SPIHT از قانون اضافه‌تری نسبت به EZW بهره می‌برد که در این قانون یک مجموعه‌ی نسل از نوع A به چهار ضریب منفرد فرزند به همراه یک مجموعه‌ی نسل بزرگ از نوع B تجزیه (یا بخش بندی) می‌شود. حال آنکه روش EZW صرفاً جستجویی از نوع Breadth First در درختهای سلسله‌مراتبی و با حرکت از زیرباندهای کلی‌تر به سمت زیرباندهای جزئی‌تر انجام می‌دهد. البته روش SPIHT نیز نوعی عمل جستجوی Breadth First انجام می‌دهد اما نه به وضوح و روشی EZW. در این روش، بعد از بخش‌بندی یک مجموعه‌ی نسل بزرگ^{۱۴۶}، چهار مجموعه‌ی نسل جدید ایجاد شده را در انتهای لیست LIS قرار می‌دهد. این عمل افزودن به انتهای لیست LIS موجب می‌شود که روش جستجو تاحدی شبیه به جستجوی Breadth First شود.

۳-۴ کدگذاری بلوکی جاسازی شده با برش بهینه شده (EBCOT)

روش کدگذاری بلوکی جاسازی شده با برش بهینه شده یا EBCOT^{۱۴۷} [۱۷] یک الگوریتم کدگذاری دیگر مبتنی بر ضرایب موجک است که قادر به گنجاندن بسیاری از امکانات و ویژگیهای پیشرفته در دنباله بیتی خود بوده و همزمان کارایی فشرده‌سازی بالایی از خود نشان می‌دهد.

با توجه به مجموعه غنی ویژگیهای این روش، پیچیدگی متوسط آن از نظر پیاده‌سازی، و کارایی فشرده‌سازی عالی این الگوریتم کدگذاری در استاندارد JPEG2000 مورد بهره برداری قرار گرفته است. قبل از بررسی این

¹⁴⁶Grand Descendant Set

¹⁴⁷Embedded Block Coding with Optimised Truncation

روش، به معایی که دو روش قبلی یعنی EZW و SPIHT داشتند و موجب عدم بهره گیری از آنها در استاندارد مذکور گردید، اشاره‌ای می‌شود. برخورداری از قابلیت‌های مقیاس پذیری مکانی و مقیاس پذیری SNR از جمله قابلیت‌های مورد نیاز و توجه در استاندارد JPEG2000 است. منظور از مقیاس پذیری مکانی این است که از روی دنباله بیتی دریافتی قادر به کدگشایی تصویر با درجات تفکیک مکانی مختلف باشیم. البته این ویژگی، مستقل از نوع روش کدگذاری ضرایب موجک، یک ویژگی ذاتی تبدیل موجک است چرا که به ازاء هر تعداد سطوح تجزیه مورد استفاده در تبدیل موجک، همواره پایینترین باند مربوطه یک نسخه تقریبی و کوچکتر از تصویر اولیه را در اختیار ما قرار می‌دهد. منظور از ویژگی مقیاس پذیری SNR نیز این است که از روی دنباله‌ی بیتی دریافتی قادر به تولید نسخه‌هایی از تصویر اولیه با کیفیت‌های مختلف باشیم. در هر دو روش EZW و SPIHT این ویژگی با انجام تکنیک تقریب‌های متوالی یا کدگذاری صفحه بیتی ضرایب موجک محقق می‌شود؛ بنابراین، به نظر می‌رسد که این دو روش قادر به تامین نیازهای مقیاس پذیری مکانی و SNR هستند اما در عمل، دنباله‌های بیتی این دو روش فقط قادر به تامین خواسته‌ی مقیاس پذیری SNR می‌باشند. در این روشها اگر بخواهیم مقیاس پذیری مکانی داشته باشیم باید ساختار دنباله بیتی را تصحیح کنیم که در این صورت، مقیاس پذیری SNR را از دست می‌دهیم زیرا ساختار درخت صفر (ZT)ی که در این روشها مورد استفاده قرار گرفته است مبتنی بر وابستگی‌های پایین‌رو^{۱۴۸} موجود بین زیرباندهای مختلف است که این خود ناشی از ساختار تجزیه متوالی موجک می‌باشد. این وابستگیها مانعی در جهت استفاده از مقیاس پذیری مکانی می‌باشند. نکته دیگر این است که وجود وابستگی بین باندهای^{۱۴۹} در ساختار درخت صفر موجب انتشار خطا در طول باندهای مختلف هم‌راستا می‌گردد که خود ناسازگار با خواسته استاندارد JPEG2000 از نظر مقاوم‌بودن در مقابل خطا است.

برای غلبه بر موانع فوق‌الذکر می‌توان هر زیرباند را به طور جداگانه کدگذاری کرد. به علاوه برای برخورداری از قابلیت انعطاف بیشتر در فرآیند کدگذاری، می‌توان هر زیرباند را به تعدادی بلوک تقسیم کرده و هر بلوک را

¹⁴⁸Downward Dependencies

¹⁴⁹Interband Dependency

جداگانه و به طور مستقل کدگذاری کرد. درون هر بلوک ممکن است وابستگی وجود داشته باشد اما بین دو بلوک مختلف این وابستگی وجود ندارد. ابعاد بلوکها وابسته به نظر کاربر در ایجاد مصالحه بین دو ویژگی کارایی کدگذاری و انعطاف در ترتیب ارسال اطلاعات در دنباله بیتی خروجی است (به عبارت دیگر در حالت کلی هر چه ابعاد بلوکها بزرگتر باشد، کارایی کدگذاری بالاتر خواهد بود اما در مقابل قابلیت انعطاف در سازماندهی دنباله‌ی بیتی خروجی و ترتیب ارسال اطلاعات کاهش می‌یابد). نکته اساسی و در واقع قلب روش EBCOT همین کدگذاری مستقل بلوکهای مختلف است. این ویژگی موجب می‌شود بتوان نمونه‌های هر بلوک را به صورت محلی پردازش کرده و در نتیجه پیاده‌سازی سخت افزاری آن راحت‌تر و موثرتر خواهد بود (چرا که پردازشها روی حجم محدود و کوچکی از اطلاعات انجام می‌شود). همچنین این روش امکان بهره‌گیری از موازی‌کاری با درجه بسیار بالا را در اختیار ما قرار می‌دهد چرا که چندین بلوک را می‌توان به طور موازی و همزمان پردازش و کدگذاری کرد. نکته بسیار مهمتر این است که به دلیل برخورداری بودن از قابلیت انعطاف زیاد در سازماندهی و ترتیب چیدن بیتها، امکان دسترسی همزمان به دو قابلیت مقیاس پذیری مکانی و SNR فراهم می‌گردد. همچنین از آنجا که خطای احتمالی مربوط به دنباله‌ی بیتی یک بلوک تاثیری روی بلوکهای دیگر ندارد، درجه بالایی از مقاوم بودن نسبت به خطا قابل حصول است. نکته پایانی این است که با توجه به عدم استفاده از شباهت و همبستگی بین زیرباند‌های مختلف، کارایی فشرده‌سازی روش EBCOT با افت مواجه خواهد بود. برای جبران این عیب، از نوع موثرتری کدگذار حسابی مبتنی بر فحوا استفاده شده است. همچنین از فرآیند بهینه‌سازی نرخ بیت-اعوجاج پس از فشرده‌سازی یا PCRD¹⁵⁰ استفاده شده است.

در روش EBCOT هر زیرباند به تعدادی بلوک نسبتاً کوچک به نام بلوک کد¹⁵¹ تقسیم می‌شود. ابعاد این بلوکها معمولاً ۳۲×۳۲ یا ۶۴×۶۴ بوده و این بلوکها مستقل از یکدیگر کدگذاری می‌شوند. الگوریتم کدگذاری EBCOT در واقع به سه مرحله قابل تجزیه است:

¹⁵⁰PostCompression Rate Distortion (PCRD) Optimization

¹⁵¹Code Block

۱- کوانتیزاسیون صفحه بیتی

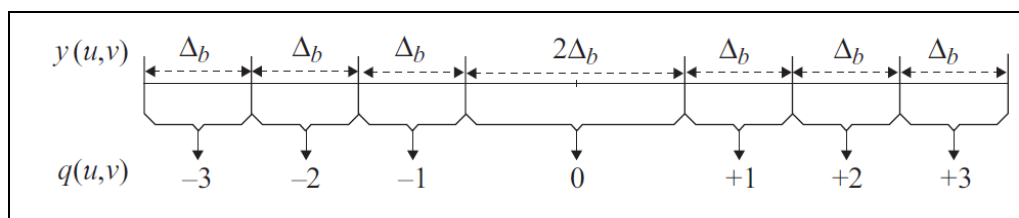
۲- کدگذاری حسابی باینری (لایه کدگذاری اول^{۱۵۲})

۳- سازمان‌دهی دنباله بیتی (لایه کدگذاری دوم)

هر کدام از سه مرحله فوق در زیربخشهای زیر توضیح داده می‌شوند.

۱-۳-۴ کوانتیزاسیون صفحه بیتی

تمام بلوکهای کد متعلق به یک زیرباند از کوانتایزر مشترکی استفاده می‌کنند. اندازه گام کوانتایزر، Δ_b ، مربوط به زیرباند مفروض b ، مبتنی بر یکی از دو عامل اهمیت ادراکی زیرباند^{۱۵۳} و یا کنترل نرخ بیت انتخاب و تعیین می‌شود. وظیفه کوانتایزر، مطابق با شکل ۴-۲، نگاشت اندازه ضریب تبدیل موجک به یک اندیس کوانتیزه شده و با حفظ علامت اولیه است. شکل مزبور، یک نوع کوانتایزر یکنواخت از نوع منطقه مرده^{۱۵۴} با اندازه گام Δ_b را نشان می‌دهد. عرض منطقه مرده دو برابر Δ_b می‌باشد.



شکل ۴-۲: کوانتایزر یکنواخت منطقه مرده با اندازه گام Δ_b

در کدگذاری صفحه بیتی، اندیسهای کوانتایزر به صورت یک بیت در هر زمان کدگذاری می‌شوند. کار ابتدا از بیت MSB شروع و به بیت LSB ختم می‌شود. اگر حداقل تعداد بیت‌های لازم برای نمایش تمام اندیسهای متعلق

¹⁵²Tier 1 Coding

¹⁵³Perceptual Importance

¹⁵⁴Uniform Dead Zone Quantiser

به یک بلوک کد برابر K باشد، آنگاه بزرگترین ضریب موجک موجود در آن بلوک کد دارای اندازه‌ای حداکثر برابر با $\Delta_b(2^k-1)$ خواهد بود. برای تولید دنباله بیتی جاسازی شده ابتدا اولین بیت MSB، یعنی بیت $(K-1)$ ام و سپس علامت تمام نمونه‌های مهم کدگذاری می‌شوند. سپس بیت MSB بعدی، یعنی بیت شماره $(K-2)$ ام کدگذاری می‌شود و به همین ترتیب تا آخرین بیت موجود در صفحه بیتی کدگذاری شود.

هرگاه دنباله بیتی از جایی به بعد قطع شود آنگاه برخی یا همه نمونه‌های موجود در بلوک فعلی ممکن است یک یا چند بیت کم‌ارزش خود را از دست بدهند. این امر معادل با استفاده از یک کوانتایزر با منطقه مرده¹⁵⁵ با اندازه گام $2^p \cdot \Delta_b$ می‌باشد که p اندیس آخرین صفحه بیت موجود و مربوط به نمونه در حال کدگذاری است.

۴-۳-۲ کدگذاری صفحه‌های بیتی با استفاده از کدگذاری حسابی شرطی

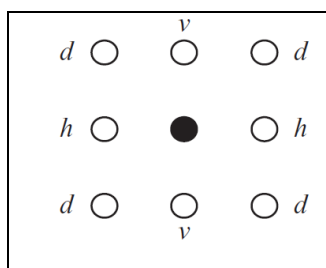
در حین فرآیند کدگذاری پیش‌رونده صفحه بیتی تزیاید قابل توجهی بین صفحات بیتی متوالی وجود دارد. الگوریتم EBCOT به دو طریق مختلف از این تزیاید استفاده کرده است. راه اول تعیین تکلیف این مطلب است که آیا ضریب موجک مفروض فعلی باید کدگذاری شود یا خیر؟ و راه دوم این است که چگونه از کدگذار آنتروپی به بهترین نحو و هماهنگ با ویژگیهای آماری ضرایب همسایه استفاده کرد؟ هر دوی این اهداف از طریق معرفی و استفاده از مفهوم وضعیت دودویی اهمیت¹⁵⁶، σ ، قابل حصول است. این مفهوم جهت مشخص کردن و شناساندن اهمیت هر ضریب در یک بلوک کد استفاده شده است. در شروع فرآیند کدگذاری وضعیت اهمیت تمامی ضرایب موجود در بلوک کد فعلی برابر صفر قرار داده می‌شوند. سپس بعد از کدگذاری اولین بیت از هر ضریب مهم این کمیت برای ضریب مذکور برابر ۱ قرار داده می‌شود. از آنجا که ضرایب همسایه معمولاً دارای وضعیت اهمیت مشابهی هستند، لذا در هر صفحه بیتی به صورت خوشه‌هایی از نمادهای دودویی ظاهر می‌شوند. بنابراین

¹⁵⁵Dead Zone

¹⁵⁶Binary Significance State

وضعیت اهمیت، σ ، برای صفحه بیت یک ضریب مفروض نشانگر خوبی است از این که آیا کدگذاری بشود یا خیر.

به علاوه در EBCOT، از کدگذار حسابی و فقی دودویی برای کدگذاری آنتروپی نمادها استفاده شده است. در اینجا هم از ویژگی خوشه بودن وضعیت اهمیت همسایگان در جهت تطابق مدل احتمالی مورد استفاده بر حسب وضعیتهای اهمیت همسایگان قبلی بهره برده می‌شود. به این نوع کدگذاری، کدگذاری حسابی دودویی مبتنی بر فحوا گفته می‌شود. در این نوع کدگذاری، احتمال لازم برای کدگذاری هر نماد 0 و 1 و یا نمادهای مثبت و منفی (مربوط به علامت یک ضریب) از روی فحوایی شامل ۸ همسایه قبلی محاسبه می‌شود. این فحوا در شکل ۳-۴ نمایش داده شده است.

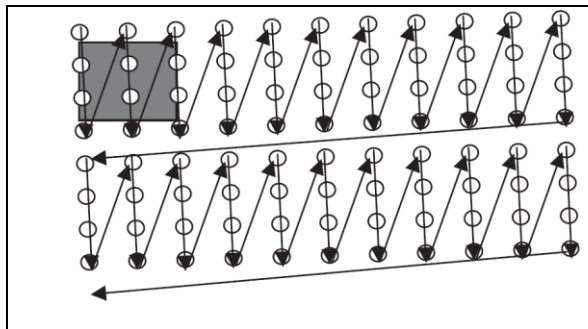


شکل ۳-۴: ۸ همسایه مورد استفاده در فحوای مورد نیاز کدگذار آنتروپی

در حالت کلی، ۸ همسایه امکان تعریف ۲۵۶ وضعیت فحوایی را فراهم می‌آورند (در هر کدام از حالت‌های استفاده از نمادهای 0/1 و یا +/-). در EBCOT تعداد مدل‌های احتمالی مورد استفاده در کدگذار حسابی محدود به ۱۸ فحوای مختلف است: ۹ فحوا برای انتشار اهمیت، ۱ فحوا برای دوره تداوم، ۵ فحوا برای کدگذاری علامت، و ۳ فحوا برای اصلاح. هر یک از فحواها در بخش‌های مربوط به خود توضیح داده می‌شود.

از آنجا که حالت و وضعیت ۸ همسایه نسبت به هم روی تشکیل هر یک از مدل‌های احتمال تاثیر دارد، بنابراین باید ضرایب هر بلوک کد را به ترتیب خاص و مشخصی جاروب و بررسی کرد. برای مثال در استاندارد JPEG2000 در هر صفحه بیت یک بلوک کد، ضرایب مطابق با شکل ۴-۴ به صورت نوار به نوار جاروب می‌شوند که عرض هر نوار برابر با ۴ پیکسل می‌باشد. در این نوع از جاروب، از گوشه سمت چپ و بالا شروع شده و چهار

بیت مربوط به اولین ستون جاروب می‌شوند. سپس ۴ بیت ستون بعدی (ستون دوم) و همین طور الی آخر تا انتهای نوار جاروب می‌شوند. حال نوبت به چهار بیت ستون اول از نوار بعدی می‌رسد و الخ. علت انتخاب ۴ به عنوان عرض (یا ارتفاع) نوار امکان پیاده سازی موثر سخت‌افزاری و نرم‌افزاری است.

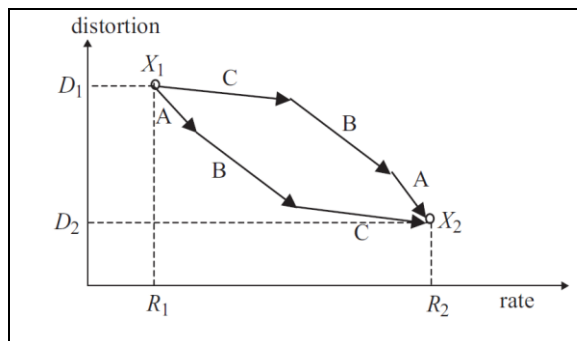


شکل ۴-۴: ترتیب جاروب نواری در یک بلوک کد مفروض

۴

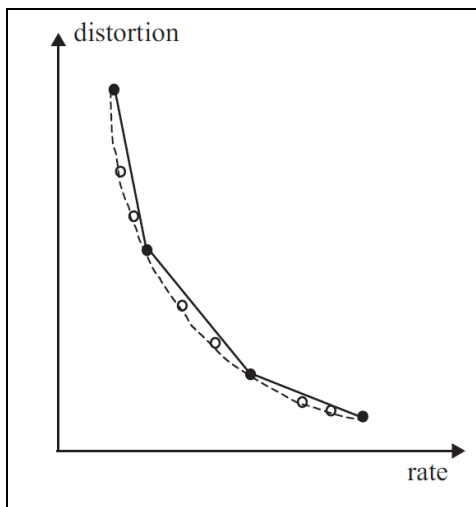
در نظر بگیرید (R_1, D_1) و (R_2, D_2) زوج نقاط نرخ-بیت-اعوجاج متناظر با دو صفحه بیتی مجاور p_1 و p_2 باشند. همچنین فرض کنید در حین هر مرحله‌ی کدگذاری (از مراحل سه گانه مذکور) نحوه‌ی افزایش نرخ بیت و کاهش میزان اعوجاج از الگو یا مسیر نمایش داده شده در شکل ۴-۵ (به کمک نمادهای A و B و C) تبعیت کنند. اگر صفحه بیت را یکباره و تماماً کدگذاری کنیم مانند این است که در شکل مزبور از نقطه X_1 به نقطه X_2 حرکت کرده‌ایم و بنابراین نوع مسیر طی شده مهم نخواهد بود چرا که در نهایت نرخ بیت از R_1 به R_2 افزایش و اعوجاج نیز از D_1 به D_2 کاهش یافته است. اما اگر قرار باشد به دلیل محدود بودن بودجه نرخ بیت^{۱۵۷} نرخ بیت را در نقطه‌ای بین R_1 و R_2 مانند $R_1 < R < R_2$ قطع کنیم آنگاه از نظر منطقی به صرفه خواهد بود که مسیر ABC را طی کنیم نه مسیر CBA.

¹⁵⁷Bit Rate Budget



شکل ۴-۵: تاثیر ترتیب کدگذاری صفحه بیتی کسری روی میزان کاهش اعوجاج

این نوع از کدگذاری صفحه بیتی کسری در واقع روشی برای بهینه‌سازی منحنی نرخ بیت-اعوجاج و به منظور تولید دنباله‌ی بیتی به خوبی جاسازی شده^{۱۵۸} بوده و در ادبیات الگوریتم EBCOT به نام بهینه‌سازی PCRD نامیده شده است [۱۸]. شکل ۴-۶ نوع بهینه‌سازی شده‌ی نرخ بیت-اعوجاج (حاصل از کدگذاری صفحه بیتی کسری) را با نوع متداول و معمول آن مقایسه می‌کند. نقاط توپر بیانگر زوج نقاط نرخ بیت-اعوجاج در پایان کدگذاری هر صفحه بیت می‌باشند. خط توپر نیز منحنی قابل انتظار حاصل از قطع دنباله بیتی در نقاط دلخواه مختلف است. از طرف دیگر در این شکل، نقاط نهایی حاصل از هر مرحله از مراحل سه‌گانه‌ی کدگذاری صفحه بیتی کسری با دایره‌های توخالی نشان داده شده است. خط شکسته نمایشگر منحنی نرخ بیت-اعوجاج حاصل از فرآیند به کارگیری این مراحل سه‌گانه و قطع آنها در هر نقطه دلخواه است. ملاحظه می‌کنید که این نقاط در پایین خط توپر (که در حقیقت نوعی درونیایی بین نقاط انتهایی توپر است) قرار گرفته‌اند زیرا مراحل اولیه کدگذاری (از مراحل سه‌گانه) دارای شیب تندی در نرخ بیت-اعوجاج می‌باشند. نتیجه‌ای که در این جا گرفته می‌شود این است که کدگذاری صفحه بیتی کسری کارایی کدگذاری تقریباً بهینه‌ای از خود در مقایسه با کدگذاری صفحه بیتی معمولی به نمایش می‌گذارد.



شکل ۴-۶: منحنی نرخ بیت-اعوجاج در حالت بهینه سازی

در حالت کلی، در کدگذاری یک ضریب، بیشترین کاهش در میزان اعوجاج زمانی رخ می‌دهد که ضریب مزبور بی‌اهمیت است اما احتمال زیادی وجود دارد که در حین کدگذاری مهم شود. مقدار متوسط کاهش در میزان اعوجاج زمانی رخ می‌دهد که قبلاً دانسته‌ایم ضریب مفروض مهم است و فرآیند کدگذاری آن را اصلاح می‌کند. و بالاخره کمترین میزان کاهش در اعوجاج زمانی رخ می‌دهد که ضریب غیرمهم باشد و بعد از کدگذاری نیز به احتمال زیادی غیرمهم باقی می‌ماند. اینها در واقع ضرایب باقی‌مانده‌ای هستند که در دو مرحله کدگذاری اولیه، کدگذاری نشده بودند. بنابراین، منطقی است که فرآیند کدگذاری صفحه بیتی را به سه مرحله تقسیم کرده و به ترتیبی که در بالا بیان شد استفاده کنیم. در JPEG2000 کدگذاری صفحه بیتی کسری در سه مرحله انجام می‌شود. نقش هر مرحله در کدگذاری و میزان اهمیت آنها در تولید دنباله بیتی بهینه در ادامه می‌آید.

۴-۳-۳-۱ مرحله انتشار اهمیت

این مرحله، اولین مرحله از کدگذاری صفحه بیتی کسری بوده و موجب تولید بیشترین میزان کاهش اعوجاج است. در این مرحله یک بیت مفروض از یک ضریب مفروض و واقع در یک صفحه بیتی فقط و فقط زمانی کدگذاری می‌شود که قبل از این مرحله وضعیت اهمیت ضریب مذکور به صورت «غیر مهم» بوده است اما حداقل یکی از ۸ همسایه این ضریب دارای وضعیت اهمیت «مهم» باشند. اگر قرار است که ضریب مذکور

کدگذاری شود نماد حاصل از بیت اندازه‌ی ضریب مذکور (یعنی 0 یا 1) کدگذاری حسابی می‌شود. در این کدگذاری از یک مدل احتمال مبتنی بر فحوایی متشکل از ۸ همسایه قبلی، مطابق با شکل ۴-۳ استفاده می‌شود. احتمال محاسبه شده برای نماد 0 متمم احتمال محاسبه شده برای نماد 1 است. انتخاب نوع فحوا با استفاده از اهمیت هر یک از ۸ همسایه مذکور انجام می‌شود که به سه دسته (یا طبقه) تقسیم می‌شود:

$$\begin{aligned}
 \text{افقی} \quad & h_i(u, v) = \sigma_i(u - 1, v) + \sigma_i(u + 1, v) \\
 \text{عمودی} \quad & v_i(u, v) = \sigma_i(u, v - 1) + \sigma_i(u, v + 1) \\
 \text{قطری} \quad & d_i(u, v) = \sum_{m=\pm 1} \sum_{n=\pm 1} \sigma_i(u + m, v + n)
 \end{aligned} \tag{۵-۴}$$

در رابطه فوق، $h_i(u, v)$ ، $v_i(u, v)$ و $d_i(u, v)$ به ترتیب همسایه‌های افقی، عمودی و قطری ضریب i ام واقع در مختصات (u, v) بوده و $\sigma_i(u, v)$ نیز بیانگر وضعیت اهمیت ضریب واقع در این مختصات می‌باشد. در محاسبه یکمیتهای فوق‌الذکر، همسایگانی که بیرون از بلوک کد قرار می‌گیرند به عنوان غیر مهم تلقی می‌شوند. به منظور بهینه سازی هزینه‌ی تطبیق‌های مدل و نیز ساده‌تر کردن پیاده‌سازی، این ۲۵۶ پیکره‌بندی مختلف همسایگان به ۹ فحوای مجزا نگاشت می‌شود که برای انجام این نگاشت از کمیتهای رابطه‌ی ۴-۵ به صورت نشان داده شده در جدول ۴-۱ استفاده می‌شود.

جدول ۴-۱: نحوه انتخاب هر یک از ۹ فحوا مبتنی بر اهمیت همسایگان

LL, LH and HL bands				HH band		
$h_i(u, v)$	$v_i(u, v)$	$d_i(u, v)$	Context	$d_i(u, v)$	$h_i(u, v) + v_i(u, v)$	Context
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	0	>1	2	0	>1	2
0	1	X	3	1	0	3
0	2	X	4	1	1	4
1	0	0	5	1	>1	5
1	0	>0	6	2	0	6
1	>0	X	7	2	>0	7
2	X	X	8	>2	X	8

در مورد باندهای HL و LH به این نکته توجه کنید که برای اینکه در مورد این باندها انتخاب فحوا به طور یکسان انجام شود، بلوکهای کد واقع در زیرباند HL ابتدا ترانهاده شده و سپس در فرآیند کدگذاری استفاده می‌شوند. زیرا زیرباند LH حساسیت و پاسخ بیشتر و قوی‌تری به لبه‌های افقی (در تصویر اصلی) می‌دهد به همین دلیل است که در جدول فوق‌الذکر که نگاشت فحواها را نشان می‌دهد، اهمیت و تاکید بیشتری روی همسایه‌های افقی شده است (از طرفی در زیرباند HL حساسیت عمده نسبت به لبه‌های عمودی است. بنابراین برای آنکه همچنان بتوان از جدول فوق‌الذکر استفاده کرد، زیرباند HL را ابتدا ترانهاده می‌کنیم تا لبه‌های عمودی متناظر با لبه‌های افقی شوند).

توجه کنید که مرحله انتشار اهمیت فقط شامل آن بیت‌هایی از ضریب مفروض می‌شود که قبل از این مرحله، غیرمهم بوده و نیز فحوایی غیرصفر داشته باشند (یعنی حداقل یکی از همسایگان مهم بوده باشد). (با توجه به لزوم برقراری این دو شرط معلوم می‌شود که در کدگذاری اولین صفحه بیت هیچگاه از مرحله انتشار اهمیت استفاده نمی‌شود زیرا در شروع کار همان طور که قبلاً نیز گفته شد، وضعیت اهمیت تمام ضرایب برابر غیرمهم مقداره‌ی اولیه می‌شود). اگر بیت فعلی از ضریب فعلی برابر ۱ باشد، آنگاه وضعیت اهمیت مربوطه، σ ، به مقدار ۱ تغییر داده می‌شود تا روی فحوای همسایگان بعدی خود تاثیر بگذارد؛ به این ترتیب، وضعیت اهمیت ضرایب در حین فرآیند کدگذاری منتشر می‌شود و به همین دلیل است که نام این مرحله از کدگذاری را انتشار/اهمیت انتخاب کرده‌اند. همچنین به این نکته توجه کنید که برای ضرایبی که در مرز یک بلوک کد قرار می‌گیرند، تنها از وضعیت اهمیت ضرایب همسایه موجود استفاده می‌شود و وضعیت اهمیت ضرایب همسایه‌ای که داخل بلوک کد قرار ندارند برابر با صفر در نظر گرفته می‌شود.

نهایتاً این که اگر معلوم شود ضریبی مهم است، علامت آن ضریب نیز کدگذاری حسابی می‌شود. از آنجا که علامت ضرایب همسایه تا حد زیادی با یکدیگر وابستگی دارند، از این وابستگی در جهت افزایش کارایی کدگذاری حسابی می‌توان استفاده کرد. برای مثال، ضرایب موجک متناظر با لبه‌های افقی و عمودی تمایل به داشتن قطبیت یکسان و مشابه با هم دارند. ضرایب قبل و بعد از لبه دارای قطبیت مخالف با هم دارند. در

الگوریتم EBCOT برای کدگذاری بیت علامت از کدگذاری حسابی با استفاده از ۵ فحواي مختلف استفاده می‌شود. انتخاب این فحواها مبتنی بر همسایه‌های افقی و عمودی ضریب فعلی است که هرکدام از اینها ممکن است یکی از این سه حالت را داشته باشند: مهم و مثبت، مهم و منفی، و غیرمهم. بنابراین چهار همسایه هر کدام سه حالت مختلف دارند و در نتیجه به تعداد $3^4 = 81$ پیکره بندی (یا ترکیب) مختلف ممکن الوقوع است. این ۸۱ حالت مختلف به ۵ فحوا نگاشت می‌شوند که جزئیات بیشتر را می‌توان در [۱۸] ملاحظه کرد.

۴-۳-۲ مرحله اصلاح اندازه

مرحله اصلاح اندازه، دومین مرحله موثر کدگذاری است. در طی این مرحله بیت اندازه ضریبی که قبلاً به عنوان یک ضریب مهم شناخته شده است، کدگذار حسابی می‌شود. این مرحله شامل بیت‌هایی است که ضریب مربوطه‌شان قبلاً به عنوان مهم شناخته شده‌اند مگر اینکه ضریبی در مرحله قبلی (که مرحله انتشار اهمیت است) به تازگی مهم شناخته شده باشد (یعنی این ضرایب را در مرحله اصلاح اندازه وارد نمی‌کنیم). در کدگذار حسابی استفاده شده از سه فحوا استفاده می‌شود که انتخاب آنها مبتنی بر مجموع وضعیت‌های اهمیت همسایگان افقی، عمودی، و قطری است. منظور از این وضعیت‌های اهمیت، آنهایی است که هم‌اکنون برای واحد کدگشا معلوم و مشخص هستند نه آنهایی که مربوط به قبل از مرحله کدگشایی اهمیت هستند. در حقیقت این نیز وابسته به این مطلب است که آیا اولین بیت اصلاح است (یعنی بیتی که بلافاصله بعد از بیت‌های اهمیت و علامت می‌آید) یا خیر.

در حالت کلی، بیت‌های اصلاح توزیع زوجی دارند مگر اینکه ضریب مفروض بلافاصله در صفحه بیت قبلی مهم شناخته شده باشد (یعنی بیت اندازه‌ای که قرار است کدگذاری شود در واقع اولین بیت اصلاح نیز باشد). این شرط در ابتدا بررسی می‌شود و در صورتی که برقرار باشد، بیت اندازه به کمک دو فحوا (که مبتنی بر وضعیت اهمیت ۸ همسایه ضریب فعلی انتخاب می‌شود) کدگذاری می‌شود (شکل ۴-۳ را ملاحظه کنید). در غیر اینصورت، بیت مذکور تنها با استفاده از یک فحوا و بدون در نظر گرفتن مقادیر همسایگان کدگذاری می‌شود.

۴-۳-۳-۳ مرحله پاکسازی

تمام بیت‌هایی که در دو مرحله قبلی (یعنی مرحله انتشار اهمیت و مرحله اصلاح اندازه) کدگذاری نشده باشند، در این مرحله (مرحله پاکسازی) کدگذاری می‌شوند. یعنی ضرایبی که غیرمهم و دارای فحوی صفر (یعنی هیچ همسایه مهمی نداشته) باشند. در حالت کلی ضرایبی که در این مرحله کدگذاری می‌شوند، ضرایبی هستند که احتمال بسیار کمی برای مهم شدن دارند و بنابراین انتظار می‌رود که غیرمهم باقی بمانند. بنابراین، از حالت خاصی به نام حالت تداوم^{۱۵۹} استفاده می‌شود. زمانی وارد حالت تداوم می‌شویم که تمام چهار ضریب (یا نمونه) واقع در یک ستون از ستونهای نوار فعلی (نواری مانند شکل ۴-۴) ضرایب غیرمهمی داشته باشند. به ویژه هرگاه شرایط زیر برقرار باشد، وارد حالت تداوم می‌شویم:

✓ چهار نمونه متوالی باید تماماً غیرمهم باشند؛ یعنی باید برای $0 \leq m \leq 3$ داشته باشیم: $\sigma_i(u+m, v) = 0$

✓ نمونه‌ها باید همسایگان غیرمهم داشته باشند؛ یعنی برای $0 \leq m \leq 3$ داشته باشیم: $h_i(u+m, v) =$

$$v_i(u+m, v) = d_i(u+m, v) = 0.$$

✓ نمونه‌ها باید همگی درون یک زیربلوک باشند.

✓ اندیس افقی اولین نمونه، یعنی u باید زوج باشد.

در حالت تداوم هر نماد باینری به کمک کدگذار حسابی و با استفاده از تنها یک فحوا کدگذاری می‌شود تا مشخص کننده این باشد که آیا تمام چهار نمونه موجود در ستون فعلی همگی غیرمهم هستند یا خیر. نماد 0 بیانگر این است که تمام نمونه‌ها صفر هستند و نماد 1 بیانگر این است که حداقل یکی از نمونه‌های ستون فعلی در صفحه بیت فعلی مهم می‌شود. هرگاه از نماد 1 استفاده شود آنگاه دو بیت دیگر نیز کدگذاری حسابی می‌شوند تا نشان دهنده مکان اولین ضریب غیرصفر در ستون فعلی باشند (با توجه به اینکه اندازه هر ستون برابر ۴ نمونه است بنابراین دو بیت کافی است). از آنجا که هریک از چهار مکان موجود در ستون فعلی بطور یکسان محتمل هستند که اولین ضریب غیرصفر در آنجا رخ دهد بنابراین از مدل احتمالی یکنواخت در کدگذار حسابی

استفاده می‌شود. بنابراین حالت تداوم تاثیر ناچیزی در کارایی کدگذاری دارد. علت اصلی استفاده از آن بهبود کارایی کدگذار حسابی از طریق تجمیع و خلاصه‌سازی^{۱۶۰} نمادها است.

بعد از مشخص کردن مکان اولین نماد غیرصفر در حالت تداوم، بقیه نمونه‌های واقع در ستون فعلی به روشی مشابه با روش انتشار اهمیت و با استفاده از همان ۹ فحوا کدگذاری می‌شوند. مشابهاً هرگاه حداقل یکی از چهار ضریب واقع در ستون فعلی دارای همسایه‌ای مهم باشد، حالت تداوم متوقف شده و مجدداً برای کدگذاری بقیه مسیر از روشی مشابه با روش انتشار اهمیت استفاده می‌شود.

برای هر بلوک کد، تعداد صفحات MSB که تماماً صفر باشند به نحوی در دنباله بیتی گنجانده می‌شود. از آنجا که وضعیت اهمیت تمام ضرایب در اولین MSB غیرصفر برابر صفر است، این صفحه فقط از مرحله پاکسازی استفاده کرده و از دو مرحله دیگر استفاده‌ای نمی‌شود.

۴-۳-۴ تشکیل لایه و سازمان‌دهی دنباله بیتی (لایه ۲ کدگذاری)

منظور از لایه ۱ کدگذاری^{۱۶۱}، انجام کدگذاری حسابی داده‌های صفحه بیتی است. این لایه برای هر بلوک کد مجموعه‌ای از دنباله‌های بیتی به همراه یک دنباله بیتی جاسازی شده‌ی مستقل تولید می‌کند. هدف لایه ۲ کدگذاری مالتی پلکش کردن دنباله‌های بیتی مذکور به منظور ارسال (به واحد کدگشا) و مشخص کردن ترتیب صفحات بیتی کدگذاری شده به شیوه‌ای موثر است. در حقیقت به خاطر استفاده از همین لایه کدگذاری است که قادر به دستیابی به ویژگیهایی مانند مقیاس ژذیری مکانی و SNR هستیم.

هر دنباله بیتی فشرده شده مربوط به بلوکهای کد مختلف، در یک یا چند لایه از دنباله بیتی فشرده شده‌ی نهایی قرار داده می‌شود. هر لایه تاثیری در جهت افزایش کیفیت خواهد داشت. تعداد مراحل که در هر لایه استفاده شده است ممکن است از یک بلوک کد به بلوک کد دیگر تغییر کند. این تعداد معمولاً توسط کدگشا و

¹⁶⁰Symbol Aggregation

¹⁶¹Tier 1 Coding

با توجه به بهینه‌سازی PCRD تعیین میشود. همان طور که قبلاً نیز گفته شد، این لایه‌های کیفیت دستیابی به قابلیت مقیاس پذیری SNR را فراهم می‌کنند به گونه‌ای که اگر دنباله‌ی بیتی نهایی را در محلی که مضرب صحیحی از لایه‌ها باشد (یعنی شامل لایه ناقصی نباشد) قطع کنیم، نمایش تقریباً بهینه‌ای از تصویر اصلی را با توجه به معیار نرخ بیت-اعوجاج فراهم می‌کند.

از آنجا که هر لایه با خود مقداری اطلاعات سربار حمل می‌کند اگر تعداد لایه‌های مورد استفاده زیاد شود با توجه به اینکه سربارها حامل اطلاعات مشترکی هستند بنابراین مزایای قابل توجهی بوجود آمده و کارایی فشرده‌سازی کاهش می‌یابد. به همین دلیل در لایه ۲ کدگذاری از این مزایا در جهت کدگذاری موثرتر اطلاعات سربار مربوط به هر لایه بهره گرفته می‌شود.

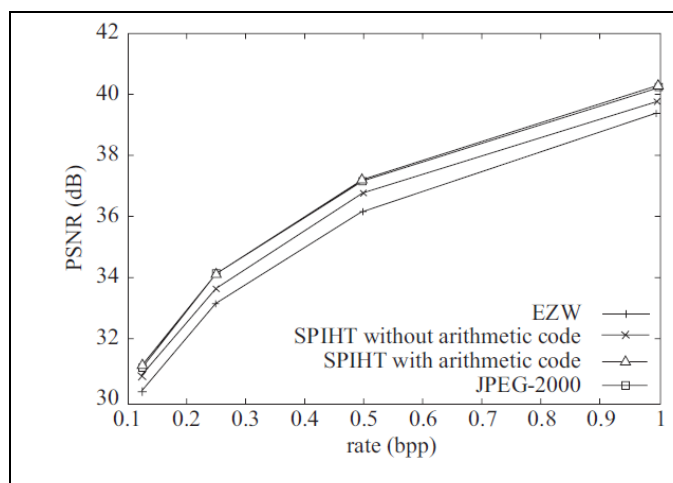
۴-۳-۵ کنترل نرخ

منظور از کنترل نرخ، فرآیند تولید یک تصویر بهینه به ازاء یک نرخ بیت مشخص بوده و مساله‌ای است که کاملاً به واحد کدگذار مربوط می‌شود. یکی از روشهای کنترل نرخ بیت، روش کدگذاری صفحه بیتی کسری است که در بخش ۴-۳-۳ معرفی شد. در مرجع [۳۳] یک روش کنترل نرخ کارا برای استفاده در الگوریتم EBCOT پیشنهاد شده است که در تنها یک گام (یا تکرار) قادر به فراهم آوردن نرخ بیت مورد نظر و با کمترین اعوجاج ممکن می‌باشد. این روش به نام بهینه‌سازی PCRD شناخته شده است. یک کدگذار JPEG2000 که گزینه‌های انتخابی متعددی در پیش رو دارد، از این روش نیز می‌تواند استفاده کند.

در شکل دیگری از PCRD هر زیرباند با استفاده از اندازه گام بسیار کوچک کوانتیزه شده و صفحات بیتی بلوکهای کد حاصل کدگذاری حسابی می‌شوند (لایه‌ی ۱ کدگذاری). این کار معمولاً مستلزم استفاده از تعداد زیادی مراحل (سه گانه‌ی) کدگذاری است که بیشتر از تعداد لازم در دنباله بیتی نهایی است. سپس به کمک بهینه‌سازی R-D لاگرانژ تعداد مراحل کدگذاری هر بلوک کد که در دنباله بیتی نهایی برای رسیدن به نرخ بیت مورد نظر مورد نیاز است تعیین می‌شود. اگر بیش از یک لایه کیفیت نیاز داشته باشیم، در پایان هر لایه

می‌توانیم این فرآیند بهینه‌سازی و تعیین تعداد مناسب مراحل را تکرار کنیم تا تعداد مراحل کدگذاری بیشتر هر بلوک کد را که در لایه بعدی استفاده کنیم، به دست آوریم. جزئیات مربوط به بهینه‌سازی PCRD در مرجع [۳۲] آورده شده است.

در پایان به مقایسه کارایی فشرده‌سازی سه روش کدگذاری ضرایب موجک یعنی EZW، SPIHT و EBCOT می‌پردازیم. شکل ۴-۷ کیفیت تصویر Lena را که به کمک این سه روش در نرخ بیت‌های مختلف کدگذاری شده است، نشان می‌دهد. همان طور که شکل مزبور نشان می‌دهد روش EZW پایین‌ترین بازدهی را دارد. SPIHT حتی بدون استفاده از کدگذاری حسابی کارایی بیشتری نسبت به روش EZW تا حد ۰.۳-۰.۴ dB دارد. اضافه کردن کدگذاری حسابی به روش SPIHT موجب افزایش کارایی فشرده‌سازی آن تا حد ۰.۳ dB می‌شود. روش EBCOT با بهترین حالت کارایی روش SPIHT برابری می‌کند.



شکل ۴-۷: بازدهی فشرده‌سازی الگوریتم‌های مختلف کدگذاری ضرایب موجک

۴-۴ مثالی از نحوه‌ی کدگذاری EBCOT

۴-۴-۱ مقدمه

تکنیک EBCOT از نوع صفحه بیتی^{۱۶۲} است به این معنا که برای کدگذاری مجموعه‌ای از ضرایب، صفحه‌ی بیتها را از با ارزشترین صفحه بیتی به کم ارزشترین صفحه بیتی کدگذاری می‌کند. در هر صفحه بیت، در حالت کلی، از سه مرحله استفاده می‌شود: انتشار اهمیت، اصلاح اندازه، و پاکسازی. همچنین از چهار واحد کدگذاری پایه^{۱۶۳} استفاده شده است: واحد دوره تداوم (^{۱۶۴}RL)، واحد کدگذاری صفر (^{۱۶۵}ZC)، واحد اصلاح اندازه (^{۱۶۶}MR)، و واحد کدگذار علامت (^{۱۶۷}SC). در این بخش به کمک یک مثال نحوه‌ی استفاده مراحل مختلف و همچنین واحد های کدگذاری پایه نشان داده می‌شود.

۴-۴-۲ مثال استفاده شده

کدگذار EBCOT، یک کدگذار بلوکی است به این معنا که ضرایب تبدیل موجک (یا به اختصار ضرایب موجک) را به صورت بلوکی کدگذاری می‌کند. در حالت کلی، بلوکها به ابعاد 64×64 می‌باشند. قسمتی از بلوک استفاده شده در این مثال در شکل ۴-۸ نشان داده شده است. فرض می‌کنیم که این بلوک متعلق به زیرباند LH می‌باشد. اعداد نشان داده شده در این شکل معرف ضرایب تبدیل موجک پس از کوانتیزاسیون می‌باشند. این ضرایب به صورت ستون به ستون جاروب و بررسی می‌شوند طوری که در هر ستون چهار ضریب قرار دارد. به عبارت دیگر ضرایب به صورت نوار به نوار مطابق با شکل ۴-۸ جاروب و بررسی می‌شوند. در این شکل، نوار فعلی به صورت مشخص نشان داده شده است. اولین سطر نشان داده شده در این شکل به نوار قبلی تعلق دارد. آخرین سطر نیز متعلق به نوار بعدی است.

در این شکل، بزرگترین ضریب برابر با ۶۱ بوده و در محل (۴،۷) قرار دارد. بنابراین صفحه‌ی بیتی متناظر با این بلوک کد فعلی، ۶ بیتی خواهد بود (زیرا $2^6 = 64$). این صفحه‌های بیتی از شماره‌ی ۰ تا ۵ شماره‌گذاری

¹⁶²Bit-Plane Coder

¹⁶³Coding Primitives

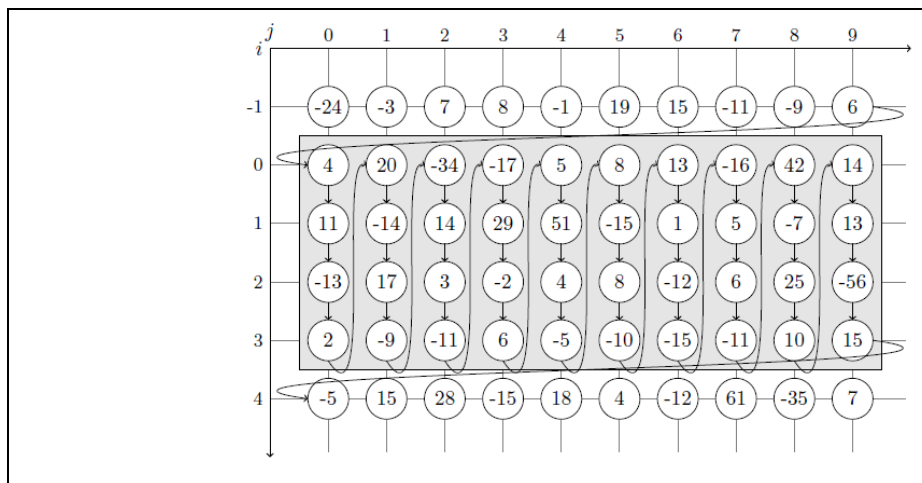
¹⁶⁴Run Length

¹⁶⁵Zero Coding

¹⁶⁶Magnitude Refinement

¹⁶⁷Sign Coding

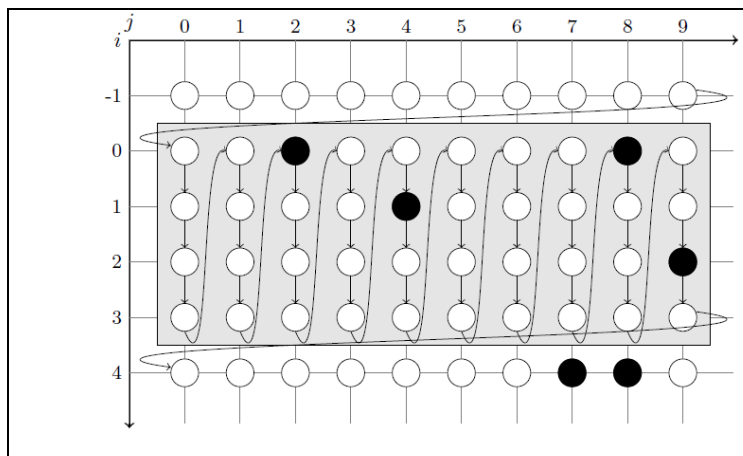
می‌شوند که عدد ۵ شماره‌ی بارزترین صفحه‌ی بیت‌ی خواهد بود. تعداد صفحات بیت‌ی هر بلوک کد، در دنباله‌ی بیت‌ی گنجانده شده و برای واحد کدگشا ارسال می‌شود تا بتواند کار خود را به طور صحیح انجام دهد.



شکل ۴-۸: بخشی از بلوک کد مورد استفاده در مثال مورد بررسی

۴-۴-۳ کدگذاری اولین صفحه‌ی بیت‌ی

اولین صفحه بیت‌ی ($n=5$) در شکل ۴-۹ نشان داده شده است. در این صفحه، تنها مرحله‌ی پاکسازی انجام می‌شود. زیرا دو مرحله‌ی دیگر قابل انجام نمی‌باشند: مرحله‌ی انتشار اهمیت به این دلیل قابل انجام نیست که هیچ ضریب موجک مهمی قبل از این صفحه‌ی بیت‌ی وجود ندارد. از آنجا که هیچ ضریب موجک مهمی تاکنون شناخته شده نیست، بنابراین نمی‌توان اهمیت را حول این ضرایب مهم، تسری داد. مرحله‌ی اصلاح اندازه نیز به دلایل مشابه قابل انجام نمی‌باشد؛ در حقیقت، برای آن که قادر به تصحیح اندازه ضرایب مهم باشیم، باید ابتدا آنها را بیابیم که در مرحله‌ی فعلی این ضرایب ناشناخته هستند. در نتیجه تنها مرحله‌ی قابل انجام در اولین صفحه‌ی بیت‌ی، مرحله‌ی پاکسازی است. این مرحله به این دلیل مرحله‌ی پاکسازی نام گرفته است که از آن به منظور کدگذاری تمام بیت‌هایی استفاده می‌شود که در دو مرحله دیگر، یعنی مرحله انتشار اهمیت و مرحله اصلاح اندازه، کدگذاری نشده‌اند.

شکل ۴-۹: صفحه‌ی بیتی $n=5$

۴-۳-۱ مرحله پاکسازی

فرآیند کدگذاری از ستون $j=0$ شروع می‌شود. قبل از کدگذاری این ستون می‌دانیم که هیچ ضریب مهمی در این ستون شناخته شده نیست. همچنین در همسایگی این ستون نیز هیچ ضریب مهمی شناخته شده نیست. هرگاه این دو شرط برقرار باشند، واحد کدگذار به وضعیت دوره تداوم^{۱۶۸} (همان واحد پایه‌ای دوره تداوم (RL) می‌رود. در این وضعیت، ستون $j=0$ جاروب می‌شود. در این ستون از صفحه‌ی بیتی $n=5$ هیچ ضریب مهمی وجود ندارد. بنابراین کدگذار که در وضعیت RL است، برای نشان دادن ستونی از صفرها، نماد 0 را تولید می‌کند. به طریق مشابه در ستون $j=1$ و همسایگی آن نیز هیچ ضریب موجه مهم شناخته شده (از قبل) وجود ندارد. بنابراین، کدگذار در وضعیت دوره تداوم باقی مانده و مجدداً نماد 0 را در وضعیت RL تولید می‌کند.

در حین ورود به ستون سوم، کدگذار همچنان در وضعیت RL قرار دارد. از آنجا که ضریب واقع در محل (۰،۲) (مطابق با شکل ۲) مهم است، واحد گذار به منظور نشان دادن به پایان وضعیت دوره تداوم (RL) نماد 1 را تولید می‌کند. به علاوه، محل ضریب مهم در ستون فعلی را نیز با تولید نماد 00 مشخص می‌کند. باین کار، واحد کدگشا می‌تواند تعیین کند که وضعیت RL دقیقاً در کدام نقطه به اتمام رسیده است. به مجرد اینکه

مشخص شد ضربی از ضرایب موجک مهم است، علامت آن کدگذاری می‌شود. بنابراین، در اینجا نیز علامت ضریب واقع در محل (۲،۰) باید کدگذاری شود.

تکنیک EBCOT برای کدگذاری علامت از پیش‌بینی مبتنی بر علامت چهار همسایه موجود استفاده می‌کند. این پیش‌بینی‌ها در جدول ۲-۴ برای سه زیرباند LL، LH، و HH نشان داده شده‌اند. نمادهای $\bar{\chi}^h$ و $\bar{\chi}^v$ بیانگر علامت دو همسایه‌ی به ترتیب افقی و عمودی می‌باشند. برای زیرباند HL این کمیتها در جدول مذکور معکوس می‌شوند.

جدول ۲-۴: علامت پیش‌بینی شده، $\hat{\chi}$ و فحوای κ^{SC} واحد پایه SC

$\bar{\chi}^h$	$\bar{\chi}^v$	κ^{SC}	$\hat{\chi}$
1	1	SC4	1
1	0	SC3	1
1	-1	SC2	1
0	1	SC1	1
0	0	SC0	1
0	-1	SC1	-1
-1	1	SC2	-1
-1	0	SC3	-1
-1	-1	SC4	-1

اگر دو همسایه مثبت باشند یا یکی از همسایه‌ها مثبت و دیگری هنوز علامتی دریافت نکرده باشد، مقدار $\bar{\chi}$ برابر ۱ می‌شود. اگر دو همسایه علامت مخالف هم داشته باشند یا اینکه هر دو علامتی هنوز دریافت نکرده باشند، مقدار $\bar{\chi}$ برابر ۰ می‌شود. اگر هر دو همسایه منفی باشند و یا اگر یکی از همسایه‌ها منفی باشد و دیگری هنوز علامتی دریافت نکرده باشد، مقدار $\bar{\chi}$ برابر ۱ می‌شود. $\hat{\chi}$ در واقع نتیجه‌ی پیش‌بینی علامت ضریب فعلی مورد کدگذاری می‌باشد. اگر این پیش‌بینی صحیح باشد، علامت 0 در فحوای SC مربوطه و در دنباله‌ی بیتی خروجی تولید می‌شود والا نماد 1 در همان فحوا تولید می‌شود.

در مورد ضریب (۲،۰) از آنجا که هیچ یک از همسایه‌ها علامتی دریافت نکرده‌اند، فحوای SC0 انتخاب می‌شود و بنابراین، علامت انتخاب شده، مثبت می‌باشد. اما این علامت صحیح نمی‌باشد بنابراین نماد 1 در فحوای SC0 در خروجی تولید می‌شود.

نمادهای (۲،۱) تا (۲،۳) در ستون $z=2$ باقی می‌مانند. از آنجا که کدگذار دیگر در وضعیت دوره تداوم (RL) نمی‌باشد، واحد پایه‌ی ZC انتخاب می‌شود. این واحد از ۹ فحوا که با ZC0 تا ZC8 مشخص شده‌اند استفاده می‌کند. این فحوها مبتنی بر اهمیت مشخص (معلوم) شده‌ی هشت همسایه‌ی ضریب کدگذاری شده‌ی فعلی بوده و در جدول ۴-۳ نمایش داده شده‌اند. نمادهای κ^h ، κ^v و κ^d به ترتیب تعداد همسایه‌های افقی، عمودی و قطری‌ای است که قبلاً به عنوان مهم شناخته شده‌اند.

جدول ۴-۳: فحوای κ^{ZC} از واحد پایه‌ی ZC. نماد x به معنای هرمقداری است.

LL and LH subbands			HL subband			HH subband		κ^{ZC}
κ^h	κ^v	κ^d	κ^h	κ^v	κ^d	κ^d	$\kappa^h + \kappa^v$	
0	0	0	0	0	0	0	0	ZC0
0	0	1	0	0	1	0	1	ZC1
0	0	≥ 2	0	0	≥ 2	0	≥ 2	ZC2
0	1	x	1	0	x	1	0	ZC3
0	2	x	2	0	x	1	1	ZC4
1	0	0	0	1	0	1	≥ 2	ZC5
1	0	≥ 1	0	1	≥ 1	2	0	ZC6
1	≥ 1	x	≥ 1	1	x	2	≥ 1	ZC7
2	x	x	x	2	x	≥ 3	x	ZC8

ضریب (۲،۱) تنها یک همسایه‌ی مهم دارد: یعنی ضریب (۲،۰). از آنجاییکه این ضریب (یعنی ضریب (۲،۱)) در صفحه بی‌تی $n=5$ مهم نمی‌باشد، بنابراین موجب تولید نماد 0 در دنباله‌ی خروجی و در فحوای ZC3 می‌شود. ضرایب (۲،۲) و (۲،۳) دارای هیچ همسایه‌ی مهمی نمی‌باشند و از طرفی در صفحه بی‌تی $n=5$ نیز مهم نمی‌باشد، بنابراین برای آنها دو نماد 0 در فحوای ZC0 تولید می‌شود.

حال واحد کدگذار به ستون $z=3$ (که شامل چهار صفر می‌باشد) می‌رسد. از آنجا که ضریب (۲،۰) در نزدیکی این ستون، مهم شناخته شده است، از وضعیت RL استفاده نمی‌شود. بلکه برای کدگذاری چهار بیت واقع در این ستون از واحد پایه‌ی ZC استفاده می‌شود. نمادهای تولید شده عبارتند از $0(ZC0)$ ، $0(ZC1)$ ، $0(ZC5)$ و $0(ZC0)$.

در ستون $j=4$ هیچ ضریب مهم شناخته شده‌ای (از قبل) وجود ندارد. به علاوه هیچ همسایه‌ی ای از این ستون مهم نمی‌باشد، بنابراین از وضعیت RL استفاده می‌شود. از آنجا که ضریب $(4,1)$ در صفحه بیتی $n=5$ مهم می‌باشد، نماد 1 در فحوای RL در خروجی تولید می‌شود. همچنین در ادامه نماد 01 در فحوای UNI تولید می‌شود تا بیانگر محل دقیق اولین ضریب مهم این ستون باشد. حال علامت این ضریب کدگذاری می‌شود. این ضریب در فحوای SC0 مثبت پیش‌بینی می‌شود. از آنجا که این پیش‌بینی صحیح می‌باشد نماد 0 در خروجی و در فحوای SC0 تولید می‌شود. دو بیت بعدی واقع در این ستون در واحد پایه‌ی ZC با نمادهای به ترتیب $0(ZC0)$ و $0(ZC3)$ کدگذاری می‌شوند.

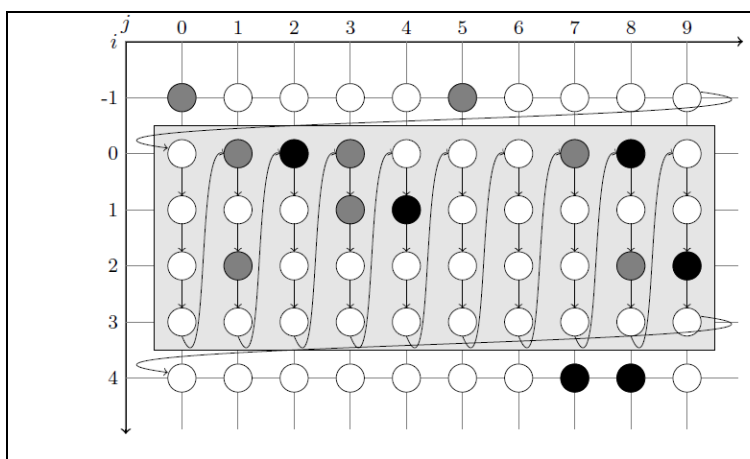
مابقی مراحل کدگذاری صفحه بیتی $n=5$ در مرحله‌ی پاکسازی در جدول ۴-۴ نشان داده شده است.

۴-۴-۴ کدگذاری دومین صفحه بیتی

دومین صفحه‌ی بیتی (یعنی $n=4$) در شکل ۴-۱۰ نشان داده شده است. در این شکل، ضرایب مهم مربوط به صفحه بیتی گذشته‌ی قبلی (یعنی صفحه بیتی مهمتر نسبت به صفحه بیتی فعلی) با رنگ سیاه نمایش داده شده‌اند. ضرایبی نیز که در صفحه بیتی فعلی مهم می‌باشند در این شکل با رنگ خاکستری نمایش داده شده‌اند. در این صفحه بیتی از هر سه مرحله انتشار اهمیت، اصلاح اندازه و مرحله پاکسازی استفاده می‌شود. ابتدا مرحله‌ی انتشار اهمیت همسایگانِ ضرایبی را که قبلاً مهم تشخیص داده شده‌اند و متعلق به صفحه بیتی فعلی یا صفحه بیتی‌های مهمتر می‌باشند را کدگذاری می‌کند. در واقع علت این که چرا ضرایب مهم را ابتدا حول بزرگترین ضرایب از قبل شناخته شده جستجو می‌کنیم این است که ضرایب با اندازه‌ی بزرگ معمولاً به صورت خوشه‌ای ظاهر می‌شوند (یعنی تمایل به تجمع در کنار یکدیگر دارند). سپس از مرحله‌ی اصلاح اندازه برای اصلاح اندازه‌ی ضرایبی که در صفحه بیتی‌های قبلاً کدگذاری شده مهم شناخته شده بودند، استفاده می‌شود. در نهایت، مرحله‌ی پاکسازی برای کدگذاری بیت‌های باقی مانده به کار گرفته می‌شود.

جدول ۴-۴: کدگذاری صفحه بیتي $n=5$ به کمک مرحله ی پاکسازی

	Coded symbols and contexts	Comments
$j = 0$	0(RL)	Run-Length.
$j = 1$	0(RL)	Run-Length.
$j = 2$	1(RL) 00(UNI) 1(SC0) 0(ZC3) 0(ZC0) 0(ZC0)	Run-Length until coefficient (0,2). Position of the first significant coefficient. Predicted sign by the context (SC0) is not correct. Remaining of the column coded by the ZC primitive.
$j = 3$	0(ZC5) 0(ZC1) 0(ZC0) 0(ZC0)	No Run-Length since coefficient (0,2) is significant.
$j = 4$	1(RL) 01(UNI) 0(SC0) 0(ZC3) 0(ZC0)	Run-Length until coefficient (1,4). Position of the first significant coefficient. Predicted sign by the context (SC0) is correct. Remaining of the column coded by the ZC primitive.
$j = 5$	0(ZC1) 0(ZC5) 0(ZC1) 0(ZC0)	No Run-Length since coefficient (1,4) is significant.
$j = 6$	0(RL)	Run-Length.
$j = 7$	0(RL)	Run-Length.
$j = 8$	1(RL) 00(UNI) 0(SC0) 0(ZC3) 0(ZC0) 0(ZC0)	Run-Length until coefficient (0,8). Position of the first significant coefficient. Predicted sign by the context (SC0) is correct. Remaining of the column coded by the ZC primitive.
$j = 9$	0(ZC5) 0(ZC1) 1(ZC0) 1(SC0) 0(ZC3)	No Run-Length since coefficient (0,8) is significant. A significant coefficient in context (ZC0). Predicted sign by the context (SC0) is not correct.



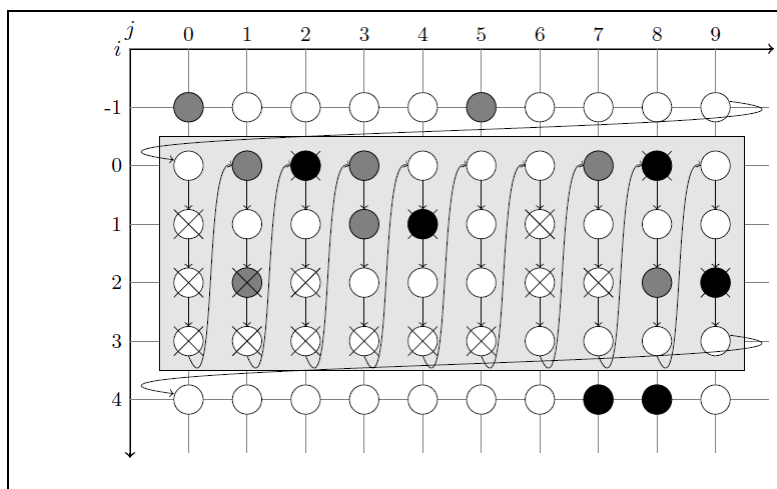
شکل ۴-۱۰: صفحه بیتي $n=4$

۴-۴-۱ مرحله انتشار اهمیت

صفحه بیتي $n=4$ از ضریب $(0,0)$ شروع می‌شود. ضریب $(-1,0)$ یکی از همسایگان این ضریب بوده و در صفحه

بیتي فعلی مهم شناخته شده است. بنابراین بیت واقع در موقعیت $(0,0)$ باید به کمک مرحله انتشار اهمیت

کدگذاری شود. در این مرحله تنها از واحد پایه ی ZC استفاده می‌شود. بنابراین نماد تولید شده در خروجی برای کدگذاری ضریب $(0,0)$ به صورت 0 و در فحوای $ZC3$ می‌باشد. ضرایب $(0,1)$ تا $(0,3)$ هیچ همسایه‌ی مهم شناخته شده‌ای (تا کنون) ندارند بنابراین در مرحله‌ی انتشار اهمیت کدگذاری نمی‌شوند. این ضرایب در شکل ۴-۱۱ با علامت ضربدر مشخص شده‌اند. از آنجا که ضریب $(2,0)$ مهم است، بیت‌های واقع در محل‌های $(1,0)$ و $(1,1)$ باید کدگذاری شوند. بیت $(1,0)$ در صفحه بی‌تی $n=4$ مهم است. بنابراین نماد $1(ZC6)$ در خروجی تولید می‌شود. حال علامت ضریب موجک به صورت $1(SC3)$ کدگذاری می‌شود چرا که علامت پیش‌بینی شده منفی است اما در واقع علامت ضریب مثبت می‌باشد. برای کدگذاری بیت $(1,1)$ نیز نماد $0(ZC3)$ تولید می‌شود. ضرایب $(1,2)$ و $(1,3)$ در این مرحله کدگذاری نمی‌شوند. ضریب $(2,0)$ نیز کدگذاری نمی‌شود زیرا این ضریب قبلاً در صفحه بی‌تی قبلی مهم تشخیص داده شده بود. برای کدگذاری بیت $(2,1)$ نماد $0(ZC3)$ تولید می‌شود. ضرایب $(2,2)$ و $(2,3)$ در این مرحله کدگذاری نمی‌شوند. مابقی اعمال انجام شده در مرحله‌ی انتشار اهمیت در جدول ۴-۵ نشان داده شده است.



شکل ۴-۱۱: مرحله‌ی انتشار اهمیت روی صفحه بی‌تی $n=4$

جدول ۴-۵: کدگذاری صفحه بی‌تی $n=4$ توسط مرحله‌ی انتشار اهمیت

	Coded symbols and contexts	Comments
$j = 0$	0(ZC3)	Coefficient (0,0) is coded.
$j = 1$	1(ZC6) 1(SC3) 0(ZC3)	Coefficient (0,1) is coded. The predicted signe in (SC3) context is not correct. Coefficient (1,1) is coded.
$j = 2$	0(ZC3)	Coefficient (1,2) is coded.
$j = 3$	1(ZC6) 0(SC3) 1(ZC7) 0(SC2) 0(ZC3)	Coefficient (0,3) is coded. Coefficient (1,3). Coefficient (2,3).
$j = 4$	0(ZC7) 0(ZC3)	Coefficients (0,4) and (2,4).
$j = 5$	0(ZC3) 0(ZC5) 0(ZC1)	Coefficients (0,5) and (2,5).
$j = 6$	0(ZC1) 0(ZC1)	Coefficients (0,6) and (3,6).
$j = 7$	1(ZC5) 1(SC3) 0(ZC3) 0(ZC3)	Coefficient (0,7). Coefficients (1,7) and (3,7).
$j = 8$	0(ZC3) 1(ZC5) 1(SC3) 0(ZC4)	Coefficient (1,8). Coefficient (2,8). Coefficient (3,8).
$j = 9$	0(ZC5) 0(ZC3) 0(ZC3)	Coefficients (0,9), (2,9), and (3,9).

۴-۴-۲ مرحله اصلاح اندازه

در مرحله‌ی اصلاح اندازه تنها از واحد پایه‌ی MR استفاده می‌شود. این واحد آن دسته از ضرایبی را که در صفحات بیت‌ی قبلی مهم شناخته شده بودند، کدگذاری می‌کند. سه فحوایی که در این واحد به کار گرفته می‌شوند در جدول ۴-۶ نشان داده شده‌اند. این فحوها با علامتهای MR0 تا MR2 مشخص شده‌اند. در این جدول، مقدار پارامتر $\tilde{\sigma}$ برابر صفر است اگر اولین بار است که واحد MR روی ضریب فعلی به کار گرفته می‌شود. در غیر این صورت برابر با ۱ است. مقدار پارامترهای k^h و k^v نیز به همان طریقی تعریف می‌شود که در واحد پایه‌ی ZC تعریف شد.

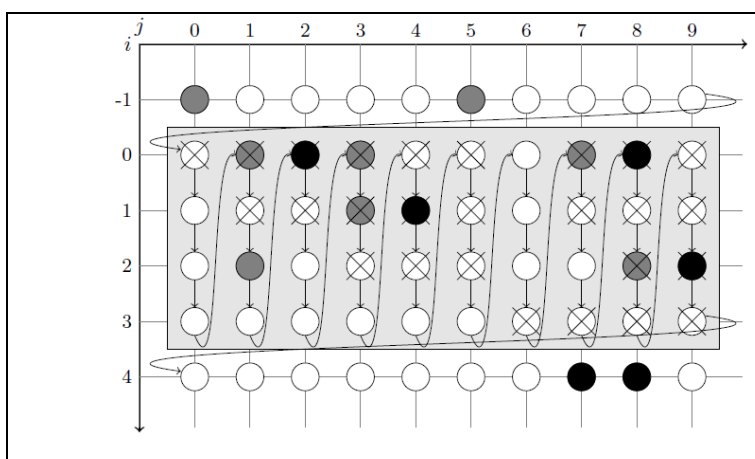
در مرحله‌ی اصلاح اندازه، اندازه ضرایبی که در صفحات بیت‌ی قبلی مهم بوده‌اند، اصلاح می‌شود. برای اصلاح اندازه‌ی ضریب (۲,۰) نماد 0(MR1) تولید می‌شود. به طریق مشابه، نمادهای 1(MR1)، 0(MR1) و 1(MR1) برای اصلاح اندازه ضرایب به ترتیب (۴,۱)، (۸,۰) و (۹,۲) تولید می‌شوند.

جدول ۴-۶: فحوهای k^{MR} مورد استفاده در واحد پایه‌ی MR

$\tilde{\sigma}$	$\kappa^h + \kappa^v$	κ^{MR}
0	0	MR0
0	$\neq 0$	MR1
1	x	MR2

۴-۴-۳ مرحله پاکسازی

از مرحله پاکسازی برای کدگذاری بیتهایی که در دو مرحله قبل کدگذاری نشده بودند استفاده می‌شود. این بیتها در شکل ۴-۱۲ با علامت ضربدر مشخص شده‌اند. اعمال انجام گرفته در این مرحله در جدول ۴-۷ نشان داده شده است.



شکل ۴-۱۲: مرحله پاکسازی روی صفحه بیتی $n=4$

۴-۴-۵ ادامه و پایان کدگذاری

فرآیند کدگذاری تا صفحه بیتی $n=0$ به صورت مشابه انجام می‌شود. از یک کدگذار حسابی MQ برای کدگذاری هر نماد خروجی با توجه به احتمال فحوای مربوطه استفاده می‌شود. مقادیر این احتمالات پس از انجام کدگذاری هر نماد به روزسانی می‌شود.

فرآیند کدگذاری بلوکهای کد که تا به حال تشریح شد، اولین بخش از کدگذار EBCOT است که به نام Tier1 شناخته می‌شود. در بخش دوم، Tier2، بخشهایی از دنباله بیتی خروجی از کدگذار صفحه بیتی، به گونه‌ای

ساماندهی می‌شوند که دنباله‌ی بیتی نهایی را از دیدگاه مصالحه‌ی نرخ-اعوجاج بهینه کند. به این فرآیند، PCRD-OPT¹⁶⁹ گفته می‌شود.

جدول ۴-۷: کدگذاری صفحه بیتی $n=4$ توسط مرحله‌ی پاکسازی

	Coded symbols and contexts	Comments
$j = 0$	0(ZC1) 0(ZC0) 0(ZC0)	
$j = 1$	1(ZC0) 0(SC0) 0(ZC3)	Coefficient (2,1).
$j = 2$	0(ZC6) 0(ZC1)	
$j = 3$	0(ZC0)	
$j = 4$	0(ZC0)	
$j = 5$	0(ZC0)	
$j = 6$	0(ZC6) 0(ZC1) 0(ZC0)	
$j = 7$	0(ZC5)	

¹⁶⁹Post-Compression Rate-Distortion OPTimization

فصل پنجم

روش پیشنهادی

۵-۱ مقدمه

روش پیشنهادی در این پایان نامه براساس تبدیل موجک، کدگذاری ضرایب تبدیل به کمک یکی از کدگذارهای بخش‌بندی مجموعه (یا SPC^{۱۷۰})، تعریف و استفاده از نواحی مطلوب در کدگذاری، و استفاده از تکنیک انطباق بلوکی به منظور کاهش تزیاید استوار است. در این فصل، پایگاه دنباله‌های ویدیویی مورد استفاده و اجزاء مختلف روش پیشنهادی توضیح داده می‌شود.

۵-۲ پایگاه داده‌ی مورد استفاده

در تهیه‌ی دنباله‌های ویدیویی ورزشی مورد استفاده در این پایان نامه، از آن جا که پایگاه داده‌ی استاندارد و قابل دسترسی در شبکه‌ی اینترنت موجود نبود، خود اقدام به تهیه‌ی چنین دنباله‌هایی نمودیم. در این کار از یک دستگاه دریافت سیگنالهای تلویزیونی^{۱۷۱} مدل Miracle (شکل ۵-۱) استفاده گردید که قادر به دریافت و ذخیره‌ی سیگنال آنالوگ (خروجی آنتن) تلویزیون و استخراج فریمها از روی آن می‌باشد. دنباله‌های ویدیویی ورزش فوتبال از روی برنامه‌های پخش مسابقات فوتبال کانالهای مختلف تلویزیونی تهیه و گردآوری گردید. حدود ۱۰ ساعت دنباله‌ی ویدیویی در پایگاه مزبور گردآوری گردیده است.

¹⁷⁰Set Partitioning Coder

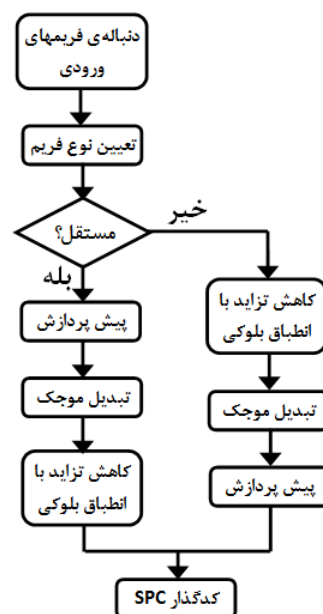
¹⁷¹DVB-T Hybrid USB TV Stick



شکل ۵-۱ کارت دریافت سیگنالهای تلویزیونی مدل Miracle نوع هیبرید (آنالوگ/دیجیتال) و نحوه‌ی اتصال آن به رایانه از طریق درگاه USB

۵-۳ روش پیشنهادی

روندنمای روش پیشنهادی در شکل ۵-۲ نشان داده شده است.



شکل ۵-۲: روندنمای روش پیشنهادی جهت فشرده‌سازی دنباله‌های ویدیویی ورزشی

مطابق با روندنمای فوق، به موازات ورود هر فریم ابتدا نوع آن از نظر مستقل یا وابسته بودن تعیین می‌شود. فریم مستقل، فریمی است که جداگانه کدگذاری می‌شود و فریم وابسته، فریمی است که با توجه به فریم کدگذاری شده‌ی قبلی کدگذاری می‌شود. هدف از استفاده از فریمهای وابسته افزایش میزان فشرده‌سازی (یا همان کارایی کدگذاری) است. در روش پیشنهادی، اولین فریم هر نما^{۱۷۲} به عنوان فریم مستقل در نظر گرفته می‌شود. روش پیشنهادی ما برای تعیین مرز نما در مرجع [۳۴] توضیح داده شده است. در زیربخش ۵-۳-۱ این روش به اختصار توضیح داده می‌شود.

معمولاً فریمهای هر نما همبستگی بسیار زیادی با یکدیگر دارند لذا استفاده از این وابستگی‌ها در جهت افزایش کارایی کدگذاری تاثیر قابل توجهی دارد. تعداد فریمهای وابسته را نمی‌توان از حدی بیشتر انتخاب کرد زیرا اگر در هنگام ارسال و کدگشایی دنباله‌ی ویدیویی، خطایی در کدگشایی یکی از فریمهای وابسته رخ دهد این خطا تا کدگشایی آخرین فریمی که به نحوی وابسته به فریم خطادار مذکور است، منتشر می‌شود. به همین دلیل، ما پیشنهاد می‌کنیم که در محیطهای ارسال نویزی، نه تنها اولین فریم هر نما را به عنوان فریم مستقل انتخاب کنیم بلکه داخل هر نما، پس از هر N1 فریم وابسته، یک فریم مستقل در نظر گرفته شود. مقدار پارامتر N1 به میزان نویزی بودن و میزان فشرده‌سازی مطلوب ما بستگی دارد. هر قدر مقدار این پارامتر بزرگتر انتخاب شود میزان فشرده‌سازی نیز تا حدی بیشتر می‌شود اما در مقابل، در محیطهای نویزی احتمال نشر خطا و کاهش کیفیت فریمهای بازسازی شده نیز بیشتر می‌شود. در ادامه، برحسب این که فریم جاری از نوع مستقل است یا وابسته یکی از دو مسیر توضیح داده شده در زیر طی می‌شود.

در مسیر اول فرض بر این است که فریم مستقل است. با توجه به روندنمای شکل ۵-۲ از یک پیش پردازش مبتنی بر تبدیل DCT به منظور مرتب کردن مولفه‌های فرکانسی و در تجمیع و انباشتن مولفه‌های هم‌مقدار در کنار یکدیگر استفاده می‌شود. این کار باعث افزایش کارایی کدگذارهای SPC (مانند SPIHT و EBCOT) می‌گردد. زیرا همان طور که در فصلهای قبلی نیز توضیح داده شده، در این کدگذارها هر قدر ضرایب

¹⁷²Shot

مهم تبدیل موجک، گروههای بزرگتر و منسجم‌تری تشکیل دهند کارایی این کدگذارها نیز بیشتر می‌شود. جزئیات روش پیش پردازش پیشنهادی در زیربخش ۵-۳-۲ آورده شده است. پس از انجام پیش پردازش، از تصویر خروجی تبدیل موجک گرفته می‌شود. در این کار، از تبدیل موجک $9/7^{۱۷۳}$ CDF استفاده می‌شود (زیرا هم دوره محدود^{۱۷۴} بوده و هم اینکه به دلیل دومتعامد^{۱۷۵} بودن دارای فاز خطی بوده و بنابراین در هنگام بازسازی سیگنال، اعوجاج نخواهیم داشت). در ادامه، به منظور افزایش کارایی کدگذاری نهایی، از انطباق بلوکی به منظور کاهش تزیاید (یا همبستگی) بین ضرایب موجک استفاده می‌شود. از آنجا که در زیرباندهای مختلف حاصل از تبدیل موجک، احتمال این که نواحی مختلف و شبیه به هم یافت شوند قابل توجه است، با استفاده از تکنیک انطباق الگو این تزیاید کاهش داده می‌شود. بنابراین، با در نظر گرفتن این که پس از انجام تکنیک مذکور، متوسط مقادیر ضرایب متعلق به بلوکهای مشابه با یک بلوک مفروض تا حد قابل توجهی کاهش یافته است، بیهیهای کمتری برای کدگذاری ضرایب متعلق به این بلوکهای مشابه لازم خواهد بود. در نتیجه، کارایی کدگذاری افزایش خواهد یافت. البته لازم به ذکر است که مختصات یکی از گوشه‌های هر بلوک مشابه یافت شده باید ذخیره شود تا در هنگام کدگذاری (بازسازی) بلوکهای ضرایب تبدیل موجک به درستی بازسازی شوند. جزئیات بیشتر روش انطباق بلوکی استفاده شده در زیربخش ۵-۳-۳ آورده می‌شود. حال، در انتهای مسیر اول و پس از اجرای تکنیک انطباق بلوکی در حوزه‌ی ضرایب تبدیل موجک، از یکی از کدگذارهای SPC استفاده می‌شود تا دنباله‌ی بیتی نهایی تشکیل شود.

در مسیر دوم فرض بر این است که فریم فعلی از نوع وابسته است. بنابراین دارای شباهت (یا تزیاید) قابل توجهی با آخرین فریم کدگذاری شده است. ابتدا با استفاده از تکنیک انطباق بلوکی در حوزه‌ی مکان (یا پیکسلی) تزیاید مزبور تا حد امکان کاهش داده می‌شود. سپس از تصویر حاصل، تبدیل موجک گرفته شده و پیش پردازش پیشنهادی در زیربخش ۵-۳-۲ روی تصویر (یا ماتریس) حاصل از تبدیل موجک اعمال می‌شود. همان طور که

¹⁷³Cohen-Daubechies-Feauveau (CDF)

¹⁷⁴Compactly-Supported

¹⁷⁵Biorthogonal

گفته شد، این پیش پردازش به منظور افزایش کارایی کدگذاری روشهای کدگذاری SPC است. در انتهای مسیر دوم از یکی از کدگذارهای SPC به منظور تولید دنباله‌ی بیتی خروجی استفاده می‌شود. در زیربخشهایی که در ادامه می‌آیند، مهمترین روشها یا تکنیکهای استفاده شده در روندنمای شکل ۵-۲ توضیح داده می‌شوند.

۵-۳-۱ تعیین نوع فریمها

آشکارسازی تغییر نما^{۱۷۶} یکی از اساسی‌ترین مراحل مورد نیاز در برخی کاربردها مانند بازیابی، مرور، اندیس‌گذاری، و استخراج قاب کلیدی است. نما معمولاً به این صورت تعریف می‌شود که شامل دنباله‌ای از فریمهای متوالی است که به طور پیوسته و مداوم از یک دوربین تهیه شده باشند یا به نظر برسد که اینگونه تهیه شده‌اند [۳۴]. انواع تغییرنما شامل دو نوع گذر ناگهانی (^{۱۷۷}AT) و گذر تدریجی (^{۱۷۸}GT) است. در گذر ناگهانی، پس از آخرین فریم نمای فعلی، بلافاصله اولین فریم بعدی ظاهر می‌شود. اما در گذر تدریجی، بین دو فریم مذکور، فریمهای دیگری قرار می‌گیرند که تغییرات را نسبتاً تدریجی از یکی به دیگری نشان می‌دهند.

در روش پیشنهادی [۳۴] برای تشخیص تغییر نما (و در نتیجه تعیین مرزنما) ابتدا از روی هر فریم ورودی (و فریم قبلی) ویژگیهایی استخراج و یک بردار ویژگی تشکیل می‌شود. با تشکیل دنباله‌ی بردارهای ویژگی و محاسبه‌ی فاصله‌ی هر دو بردار متوالی، دنباله‌ی فاصله‌ها به دست می‌آید. با بررسی این دنباله و مقایسه‌ی آن با یک آستانه هنگام تغییر مرزنما آشکارسازی می‌شود. ویژگیهایی که در این روش پیشنهادی استفاده شده‌اند عبارتند از میانگین، واریانس، فاصله‌ی هیستوگرامها، مرکز ثقل و پهنای باند مولفه‌های رنگ، و قدرت لبه‌ها [۳۴].

۵-۳-۲ پیش پردازش پیشنهادی

¹⁷⁶ Shot

¹⁷⁷ Abrupt Transition (AT)

¹⁷⁸ Gradual Transition (GT)

پیش پردازش پیشنهادی هم در حوزه‌ی مکان (پیکسلی) و هم در حوزه‌ی تبدیل قابل استفاده است. بنابراین، ورودی ممکن است ماتریس تصویر و یا ماتریس ضرایب تبدیل باشد. اگر ورودی، ماتریس تصویر باشد ابتدا از آن تبدیل DCT گرفته و پس از اعمال پیش پردازش پیشنهادی، از نتیجه، عکس تبدیل DCT می‌گیریم. اما اگر ورودی ماتریس ضرایب تبدیل باشد بدون هیچ تغییری، پیش پردازش پیشنهادی را روی آن اعمال می‌کنیم. در هر صورت ما در این جا برای راحتی، ماتریس ورودی را تصویر ورودی می‌نامیم.

برای انجام پیش پردازش روی تصویر ورودی، ابتدا آن را به بلوکهای مربعی به اندازه‌ی $N2$ تقسیم کرده و برای هر بلوک متوسط مقادیر متعلق به آن بلوک را محاسبه می‌کنیم. از روی این مقادیر متوسط، دنباله‌ای تشکیل می‌دهیم که هر مقدار این دنباله متناظر با بلوکی از بلوکهای تصویر ورودی است. حال این دنباله را به ترتیب نزولی مرتب کرده و دنباله‌ی جدیدی به دست می‌آوریم. سپس محل بلوکهای متناظر با مقادیر دنباله‌ی مرتب شده را به ترتیب زیگزاگ از گوشه‌ی سمت چپ و بالای تصویر ورودی به سمت گوشه‌ی سمت راست و پایین تصویر ورودی تغییر می‌دهیم. برای مثال، اگر اولین مقدار متعلق به دنباله‌ی جدید، پنجمین مقدار متعلق به دنباله‌ی اولیه باشد، باید مطابق با شکل ۳-۵ جای این بلوک را در تصویر خروجی تغییر دهیم. ترتیب زیگزاگ در هر دو شکل ۳-۵ الف و ۳-۵ ب با اعداد مشخص شده است.

1	2	6	7	15
3	5	8	14	16
4	9	13	17	24
10	12	18	23	25
11	19	22	26	29
20	21	27	28	30

1	2	6	7	15
3	5	8	14	16
4	9	13	17	24
10	12	18	23	25
11	19	22	26	29
20	21	27	28	30

شکل ۳-۵: مرتب کردن بلوکها به ترتیب نزولی، تغییر مکان بلوکهای اولیه و تشکیل تصویر جدید

اگر تصویر ورودی همان تصویر اصلی (در حوزه مکان یا پیکسل) باشد، از آنجا که ابتدا از تصویر تبدیل DCT گرفته شده و سپس پیش پردازش پیشنهادی روی آن اعمال می‌شود، نتیجه این خواهد بود که تصویری با طیف عمدتاً فرکانس پایین به دست خواهد آمد؛ یعنی تصویری که دارای تغییراتی نرمتر از تصویر اصلی است. زیرا هر قدر سیگنال مولفه‌های فرکانس پایین غالبتر داشته باشد، در حوزه‌ی زمان تغییرات نرمتری از خود نشان خواهد داد. برای مثال شکل ۴-۵-الف یک تصویر اصلی و شکل ۴-۵-ب نتیجه‌ی اعمال فرآیند گفته شده (یعنی تبدیل DCT، اعمال پیش پردازش، و سپس تبدیل عکس DCT) را نشان می‌دهد. ملاحظه می‌کنید که تصویر نتیجه دارای تغییرات بسیار نرمتری نسبت به تصویر اولیه است. بنابراین، در ماتریس تبدیل موجک چنین تصویر همواری، ضرایب بزرگ تمایل به تشکیل خوشه‌های بزرگ و منسجم داشته و در نتیجه، کارایی کدگذارهای SPC افزایش خواهد یافت.



ب

الف

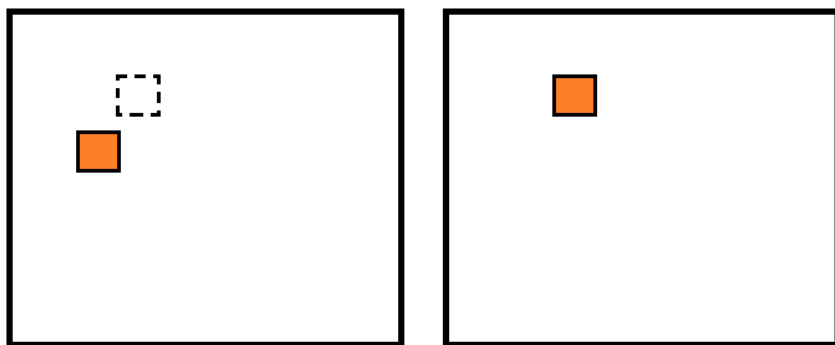
شکل ۴-۵: (الف) تصویر اصلی، (ب) نتیجه‌ی فرآیند پیش پردازشی پیشنهادی

از طرف دیگر اگر تصویر ورودی، ماتریس ضرایب تبدیل باشد، پیش پردازش پیشنهادی باعث منسجم‌تر شدن و انباشته شدن ضرایب تقریباً هم‌اندازه در کنار یکدیگر می‌شود که مجدداً موجب افزایش کارایی کدگذاری روشهای کدگذاری SPC می‌گردد.

شماره‌ی اولیه‌ی بلوکها به هنگام مرتب شدن زیگزاگی باید در جایی ذخیره و برای واحد کدگشا ارسال شود تا در هنگام بازسازی تصویر اصلی مورد استفاده قرار گیرد.

۵-۳-۳ کاهش تزايد به کمک انطباق بلوكی

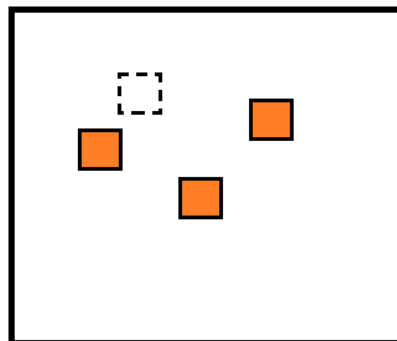
تکنیک انطباق بلوک‌ی روشی برای پیدا کردن بلوکهای مشابه در تصویر ورودی است. دو بلوک را مشابه می‌گوییم هرگاه مقدار اختلاف آنها کمتر از آستانه‌ی از قبل مشخصی مانند T_{BM} باشد. مقدار اختلاف دو بلوک در این پایان نامه به کمک معیار MSE محاسبه می‌شود. مقدار آستانه T_{BM} به کمک آزمایش تعیین می‌شود. در حالت کلی تکنیک انطباق الگو روشی جهت یافتن بلوکهای مشابه با هم در یک یا دو تصویر مختلف است. هرگاه دو تصویر مختلف در دست باشد و بخواهیم این تکنیک را بین آن دو اجرا کنیم، منظور این است: اول این که دو تصویر یکی فریم فعلی و دیگری فریم قبلی است. دوم این که برای هر بلوک متعلق به یک تصویر (بلوک اولیه)، شبیه‌ترین بلوک به آن بلوک ولیواقه در تصویر دیگر (بلوک مشابه) را طوری پیدا کنیم که بلوک مشابه یافت شده در یک همسایگی متناظر با بلوک اولیه قرار داشته باشد. شکل ۵-۵ سعی در نمایش این هدف دارد. در شکل ۵-۵-الف بلوک اولیه به صورت توپُر نمایش داده شده است. در شکل ۵-۵-ب بلوک اولیه به صورت خط چین و بلوک مشابه یافت شده به صورت توپُر نمایش داده شده است.



ب

الف

شکل ۵-۵: انطباق بلوک‌ی در حالت دو تصویر ورودی، (الف) فریم فعلی، (ب) فریم قبلی



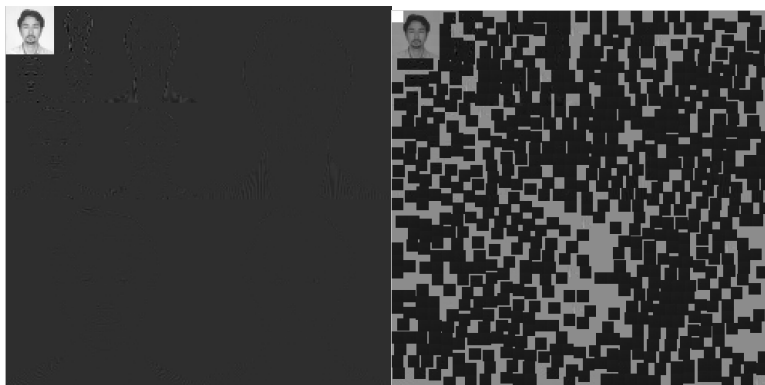
شکل ۵-۶: انطباق بلوکی در حالت یک تصویر ورودی؛ در این حالت تصویر ورودی همان ماتریس ضرایب تبدیل موجک است. هدف این است که برای یک بلوک مفروض (مانند بلوک خط چین) تمام بلوکهای مشابه و واقع در یک همسایگی (مانند بلوکهای خط ممتد) یافت شوند.

در حالتی که تکنیک انطباق الگو را بخواهیم در مورد یک تصویر ورودی اعمال کنیم، منظور این است که: اول این که آن تصویر، ماتریس ضرایب تبدیل موجک است. دوم این که منظور این است که برای یک بلوک مفروض (بلوک اولیه) تمام بلوکهای مشابه و واقع در یک همسایگی را پیدا کنیم (بلوکهای مشابه). این کار در شکل ۵-۶ نمایش داده شده است.

در روش پیشنهادی (روندنمای شکل ۵-۲) از تکنیک انطباق الگو در دو حوزه‌ی مختلف مکانی (یا پیکسلی) و تبدیل استفاده می‌شود. به عبارت بهتر، در صورتی که فریم جاری از نوع مستقل باشد، از تکنیک انطباق الگو در حوزه‌ی تبدیل و در غیر این صورت از تکنیک مذکور در حوزه‌ی مکانی استفاده می‌شود. در هر حال، در این جا برای راحتی، ورودی این تکنیک را تصویر (یا تصویرهای) ورودی می‌نامیم. اگر فریم جاری، مستقل باشد منظور از تصویر ورودی، ماتریس ضرایب تبدیل موجک و اگر فریم جاری، وابسته باشد منظور از تصویرهای ورودی، همان فریم فعلی و فریم قبلی است.

تکنیک انطباق بلوکی به منظور کاهش تزايد مکانی در حوزه ضرایب تبدیل موجک (حالت اول) و یا حوزه مکانی (حالت دوم) استفاده شده است. با این کار، در حالت اول، اندازه‌ی ضرایب تبدیل موجک به طور متوسط کاهش یافته و در نتیجه، تعداد بیت‌های لازم برای کدگذاری آنها (در روش‌های SPC مانند SPIHT و EBCOT) کاهش می‌یابد. همچنین، این روش موجب تولید برخی بلوک‌های کد تمام-صفر در برخی زیرباند‌های تبدیل موجک می‌شود. روش کدگذاری EBCOT نسبت به روش کدگذاری SPIHT قدرت بیشتری برای بهره گرفتن از این بلوک‌های کد تمام صفر دارد. بنابراین، انتظار داریم این تکنیک در مورد کدگذار EBCOT موثرتر واقع شود. در حالت دوم نیز تکنیک انطباق بلوکی، شباهت بین بلوک‌های متعلق به دو فریم فعلی و قبلی را بررسی و بلوک‌های مشابه را آشکارسازی می‌کند. سپس به جای هر بلوک مشابه یافت شده در فریم فعلی، تفاضل آن با بلوک مشابه متناظر در فریم قبلی قرار داده می‌شود. به این ترتیب، مجدداً متوسط اندازه‌ی مقادیر پیکسل‌های بلوک متعلق به فریم فعلی کاهش یافته و در نهایت، منجر به کاهش تعداد بیت‌های لازم جهت کدگذاری آنها می‌گردد.

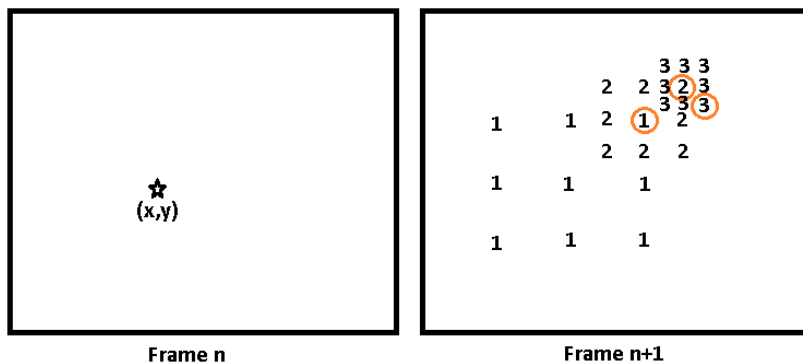
اگر تکنیک انطباق بلوکی را در حالت اول یعنی تصویر حاصل از ضرایب تبدیل موجک بخواهیم استفاده کنیم، روش کار در تکنیک پیشنهادی به این صورت است که تصویر ورودی به بلوک‌های مربعی با ابعاد از قبل مشخص BMM تقسیم می‌شود. این بلوک‌ها در حال کلی می‌توانند با یکدیگر درصد همپوشانی OV هم در راستای سطری و هم در راستای ستونی داشته باشند. معمولاً این مقدار برابر ۵۰٪ انتخاب می‌شود. حال برای هر یک از این بلوک‌ها، تمام دیگر بلوک‌ها بررسی شده و بلوک‌هایی که شرط مشابه بودن را برآورده کنند شناسایی می‌شوند. سپس به جای چنین بلوک‌هایی، تفاضل آنها از بلوک اولیه متناظر جایگذاری می‌شود. البته برای هر یک از چنین بلوک‌هایی، مختصات نقطه بالا سمت چپ نیز ذخیره می‌شود تا به واحد کدگشا در بازسازی ماتریس اولیه تبدیل موجک کمک کند. نکته‌ی مهم در تکنیک انطباق بلوکی پیشنهادی این است که هرگاه دو بلوک با یکدیگر مشابه تشخیص داده شدند، از آن به بعد هیچ بلوک یا بلوک‌های مشابه دیگری نباید با این دو بلوک همپوشانی داشته باشند زیرا کار واحد کدگشا در بازسازی بلوک‌های اولیه را دچار مشکل می‌کنند. شکل ۵-۷ نمونه‌ای از چنین انطباق الگویی در حوزه‌ی تبدیل موجک را نشان می‌دهد.



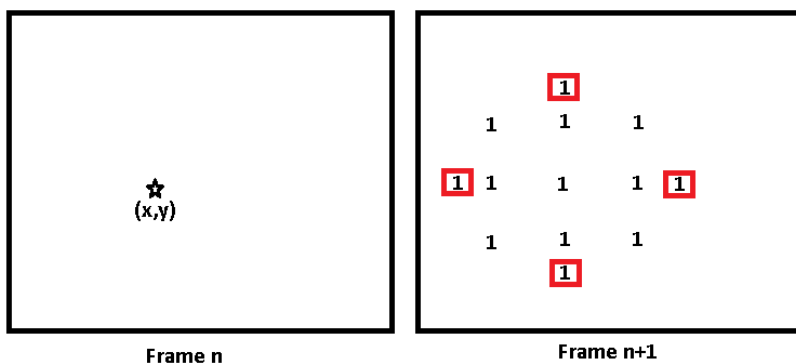
الف ب

شکل ۵-۷: یک نمونه ماتریس تبدیل موجک (الف) و نتیجه انطباق بلوکی (ب) روی ماتریس (الف)

اگر تکنیک انطباق بلوکی را در حالت دوم یعنی حوزه مکانی و استفاده از فریم فعلی و فریم قبلی استفاده کنیم، روش کار در تکنیک پیشنهادی به این صورت است که فریم فعلی و فریم قبلی (که ممکن است خود یک فریم وابسته و یا فریم مستقل باشد) به تعدادی بلوک مربعی با ابعاد از قبل مشخص B_{BM} تقسیم شده و برای هر بلوک متعلق به فریم فعلی، شبیه ترین بلوک در فریم قبلی جستجو می‌شود. برای انجام جستجو تعدادی بلوک کاندید بررسی می‌شوند. این که چه بلوک‌هایی بررسی شوند به نوع روش انطباق بلوکی بستگی دارد. ما در کار خود دو نوع جستجو را پیشنهاد و استفاده می‌کنیم. نوع اول جستجو برای بلوک‌های متعلق به نواحی مطلوب مورد استفاده قرار می‌گیرد زیرا دارای دقت بیشتری نسبت به نوع دوم است. نوع دوم نیز برای بلوک‌های معمولی استفاده می‌شود. در نوع اول که جستجوی سه مرحله‌ای نام دارد [۳۵ و ۳۶]، از بلوک‌های کاندیدی که مرکز آنها در شکل ۵-۸ نشان داده شده است استفاده می‌شود. در نوع دوم نیز که جستجوی تک مرحله‌ای نام دارد و در این پایان نامه پیشنهاد شده است، از الگوی بلوک‌های مشخص شده در شکل ۵-۹ استفاده می‌شود.



شکل ۵-۸: الگوی بلوکی روش جستجوی سه مرحله‌ای



شکل ۵-۹: الگوی بلوکی روش جستجوی پیشنهادی تک مرحله‌ای

نکته‌ی مهمی که در تکنیک انطباق بلوکی باید در نظر گرفت این است که برای بررسی شباهت داشتن یا نداشتن دو بلوک مفروض باید از یک آستانه استفاده شود. مقدار این آستانه باید با توجه به برقراری یک مصالحه بین یک بده-بستان^{۱۷۹} انتخاب شود. این بده-بستان بین دو پارامتر تعداد بلوکهای مشابه یافت شده و انرژی بلوکهای باقیمانده وجود دارد. می‌دانیم که هر قدر مقدار آستانه را کوچکتر انتخاب کنیم، میزان شباهت محاسبه شده با آستانه‌ی کمتری مقایسه شده و احتمال این که دو بلوک، یکسان و مشابه در نظر گرفته شوند بیشتر می‌شود. هر قدر تعداد بلوکهای مشابه یافت شده بیشتر شود، به ظاهر، میزان تزیاید (همبستگی) بیشتری حذف می‌شود. اما از طرف دیگر و در مقابل، بلوکهای باقیمانده‌ای که حاصل خواهند شد دارای انرژی (واریانس) بیشتری هستند که خود مستلزم صرف بیهوشی بیشتر به منظور کدگذاری این بلوکهای باقیمانده

179 Trade-off

است. مصالحه بین این دو پارامتر باید در جهت بیشینه کردن میزان فشرده‌سازی انجام شود که این کار، در این پایان نامه، به صورت تجربی انجام شده است.

۵-۳-۴ تعیین نواحی مطلوب

روش فشرده‌سازی دنباله‌های ویدیویی ورزش فوتبال که در این پایان نامه پیشنهاد شده است، مبتنی بر نواحی مطلوب است؛ به این معنا که در حالت کلی، در هر فریم برخی نواحی از اهمیت بیشتری نسبت به دیگر نواحی برخوردار هستند و بنابراین در کاربردهای نرخ بیت پایین، بهتر است این نواحی مطلوب با نرخ بیت بیشتری نسبت به دیگر نواحی کدگذاری شوند. بنابراین، اولین قدم تعیین نواحی مطلوب و دومین قدم، کدگذاری وفقی با توجه به نوع ناحیه‌ی درحال کدگذاری است. یکی از محاسن کدگذارهای SPC این است که در آنها مقدار بیت مصرفی برای هر ناحیه قابل تعیین است.

در ویدیوهای ورزش فوتبال نواحی مختلفی را می‌توان ناحیه‌ی مطلوب نامید. ما در روش پیشنهادی خود، دو نوع ناحیه‌ی مطلوب تعریف و استفاده کرده‌ایم. نوع یا سطح اول، آن نواحی‌ای که دارای تغییر و حرکت باشند را به عنوان نواحی مطلوب در نظر گرفته‌ایم که عمدتاً متناظر با حرکت توپ و بازیکنان می‌باشند. در سطح دوم، ناحیه‌ی زمین فوتبال را نیز به ناحیه‌ی قبلی افزوده‌ایم. برای آشکارسازی نواحی نوع اول از تفریق فریمها استفاده می‌کنیم. در شکل ۵-۱۰ دو فریم متوالی و حاصل تفریق آنها نشان داده شده است. در تصویر تفریق، برای راحتی مشاهده، نواحی سیاه رنگ، متناظر با حرکت اشیا و یا دوربین می‌باشند.

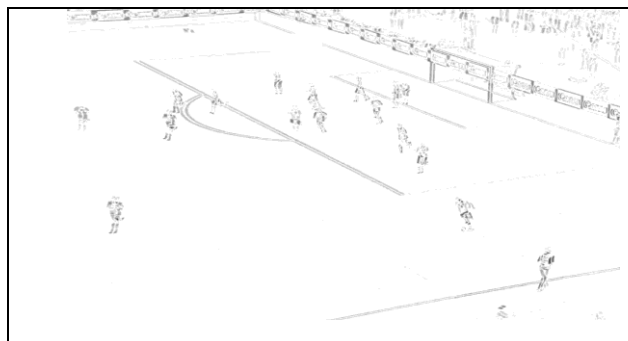
برای شناسایی نواحی مطلوب سطح دوم، از خوشه‌یابی هر فریم مبتنی بر ویژگیهای رنگ سبز و فاصله مولفه رنگ سبز از دو مولفه‌ی رنگ دیگر استفاده شده است. بنابراین، بردار ویژگی مورد استفاده در کار ما شامل سه مولفه‌ی G ، $|R-G|$ ، و $|B-G|$ است. خوشه‌یابی به کمک روش K-means و با سه خوشه انجام می‌شود. بزرگترین خوشه از نظر مساحت اشغالی در هر فریم، به شرطی که در ادامه گفته خواهد شد، به عنوان ناحیه‌ی زمین چمن

در نظر گرفته می‌شود. شکل ۵-۱۱ یک نمونه خوشه‌یابی را نشان می‌دهد که بزرگترین خوشه در آن با رنگ سفید و بقیه‌ی دو خوشه با رنگ سیاه نمایش داده شده‌اند.



ب

الف



ج

شکل ۵-۱۰ (الف و ب) دو فریم متوالی، (ج) تصویر حاصل از تفریق دو فریم (الف) و (ب)



شکل ۵-۱۱ یک نمونه نتیجه‌ی انجام خوشه‌یابی با سه خوشه و انتخاب بزرگترین خوشه به عنوان ناحیه‌ی زمین چمن

در صورتی که هیچ یک از سه خوشه‌ی شناسایی شده به صورت یک ناحیه‌ی بزرگ که (حداقل) در یکی از دو راستای فریم حداقل ۵۰ درصد گسترش یافته باشد، ظاهر نشود، ناحیه‌ی مطلوب از نوع دوم نخواهیم داشت. در این حالت در فریم ویدیویی، احتمالاً زمین فوتبال (به طور غالب و واضح) مشاهده نمی‌شود.

۵-۳-۵ دیگر حالت‌های کاری روش پیشنهادی

روش پیشنهادی برای رسیدن به نرخ بیت‌های پایین‌تر دارای سه حالت کاری دیگر نیز می‌باشد که هر کدام به تنهایی یا با هم قابل استفاده‌اند. این حالت‌های کاری، کمک نسبتاً زیادی به کاهش نرخ بیت می‌کنند. حالت کاری اول شامل کاهش نرخ نمونه برداری زمانی و به عبارت دیگر کاهش تعداد فریمها در واحد زمان است. ویدیوها معمولاً با نرخ ۲۵ الی ۳۰ فریم بر ثانیه تولید می‌شوند. در این حالت ما از هر دو فریم متوالی، اطلاعات یکی را حذف کرده و کدگذاری نمی‌کنیم. با این کار نرخ فریمها و بنابراین حجم داده‌ها در واحد زمان به نصف تقلیل می‌یابد. در موقع بازسازی، برای رسیدن به نرخ فریم اولیه، از هر دو فریم متوالی یک فریم را به روش درونیابی خطی تولید و بین آن دو فریم مربوطه جایگذاری می‌کنیم تا به همان نرخ فریم اولیه برسیم.

در حالت کاری دوم، نرخ نمونه برداری مکانی هر فریم را کاهش می‌دهیم. برای این کار، در هر سطر و هر ستون از یک فریم، از هر دو پیکسل یکی را حذف می‌کنیم. به این ترتیب ابعاد هر فریم به نصف ابعاد اولیه تقلیل می‌یابد. در هنگام بازسازی، برای رسیدن به همان نرخ نمونه برداری مکانی اولیه، بین هر دو پیکسل مجاور، یک پیکسل جدید از طریق درونیابی خطی محاسبه و جایگذاری می‌کنیم.

حالت کاری سوم روش پیشنهادی، کاهش دقت بیتی استفاده شده برای نمایش مقادیر مولفه‌های رنگ پیکسلها است. برای نمایش هر پیکسل، معمولاً از ۸ بیت برای نمایش هر یک از مولفه‌های رنگ استفاده می‌شود. در این حالت کاری، از روش کوانتیزاسیون ساده (یکنواخت) برای کاهش تعداد رنگ هر پیکسل استفاده می‌کنیم تا دقت بیتی برای نمایش هر مولفه‌ی رنگ از ۸ بیت به ۶ بیت کاهش یابد.

۵-۴ نتیجه‌گیری

در این فصل روش پیشنهادی برای فشرده‌سازی دنباله‌های ویدیویی ورزش فوتبال توضیح داده شد. در روش پیشنهادی هر فریم ابتدا به یکی از دو نوع مستقل و وابسته دسته‌بندی می‌شود. تعیین نوع فریم به کمک روش پیشنهادی برای تشخیص تغییر نما انجام می‌شود. سپس برحسب این که فریم فعلی مستقل یا وابسته است یکی از دو مسیر مشخص شده پی گرفته می‌شود. اگر فریم فعلی وابسته باشد ابتدا تزايد مکانی آن به کمک تکنیک انطباق بلوکی پیشنهادی کاهش داده شده و سپس از آن تبدیل موجک گرفته می‌شود. در ادامه از پیش پردازش پیشنهادی به منظور افزایش کارایی کدگذاری کدگذار SPC استفاده شده (مانند SPIHT و EBCOT) استفاده شده و در انتها از کدگذار مذکور به منظور تولید دنباله‌ی بیتی خروجی استفاده می‌شود.

اگر فریم فعلی از نوع مستقل باشد، ابتدا به کمک پیش پردازش پیشنهادی، تبدیل به تصویری هموار (که موجب افزایش کارایی کدگذاری کدگذار SPC است) می‌شود؛ سپس، از آن تبدیل موجک گرفته شده و در حوزه‌ی ماتریس ضرایب موجک، از تکنیک انطباق بلوکی پیشنهادی به منظور کاهش تزايد بین ضرایب مختلف استفاده می‌شود. در انتها از کدگذار SPC به منظور تولید دنباله‌ی بیتی خروجی استفاده می‌شود.

نوآوریهای قابل طرح در این پایان نامه عبارتند از: استفاده از تشخیص تغییر نما در کاربرد فشرده‌سازی و ارائه روش پیشنهادی تشخیص تغییر نما جهت افزایش کارایی کدگذاری فریمهای وابسته، انطباق بلوکی و استفاده از آن در دو جهت کاهش تزايد فریمهای متوالی (و در نتیجه افزایش کارایی فشرده‌سازی) و کاهش تزايد درونی (در حوزه‌ی ماتریس ضرایب تبدیل)، تعریف و شناسایی نواحی مطلوب در صحنه ورزش فوتبال، استفاده از کدگذاری نواحی مطلوب با استفاده از کدگذارهای SPC (به ویژه کدگذار EBCOT)، و تعریف بلوکهای کاندید جدید در انطباق بلوکی به منظور انجام انطباق بلوکی افقی (با توجه به مطلوب بودن/نبودن ناحیه‌ها از الگوهای بلوکی مختلف استفاده می‌شود).

فصل ششم

نتایج شبیه سازی

۶-۱ مقدمه

روش پیشنهادی در این پایان نامه براساس تبدیل موجک، کدگذاری ضرایب آن به کمک کدگذارهای SPC و مبتنی بر نواحی مطلوب، و استفاده از برخی تکنیکها مانند انطباق بلوکی (در حوزه‌های مختلف)، بازچینی بلوکها، کاهش نرخ نمونه برداری زمانی، کاهش نرخ نمونه برداری مکانی، و کاهش نرخ نمونه برداری مقدار استوار است. در این فصل اثر این اجزاء روی کارایی فشرده سازی در قالب ترکیبات خاصی بررسی می شود.

۶-۲ بررسی میزان تاثیر تکنیکهای انطباق بلوکی و بازچینی بلوکها روی کارایی کدگذارهای SPC

در روش پیشنهادی، از تکنیک انطباق بلوک در دو حوزه‌ی تبدیل (برای فریمهای مستقل) و مکانی (برای فریمهای وابسته) استفاده می شود. هدف از این تکنیک، بالابردن کارایی کدگذاری کدگذارهای SPC است زیرا هر قدر بلوکهای صفر (یا با متوسط اندازه‌ی ضرایب کوچک) به ویژه در جهت‌های متناظر در زیرباندهای حاصل از تبدیل موجک بیشتر شود، کارایی فشرده سازی این کدگذارها بیشتر می شود. قابل یادآوری است که یکی از ویژگیها یا فرضیه‌هایی که این کدگذارها بر اساس آن کار می کنند این است که اگر مجموعه‌ای از ضرایب در بالاترین باندهای تبدیل موجک کوچک و یا صفر باشند، به احتمال بسیار زیاد تمام ضرایب متناظر با این مجموعه و واقع در تمام زیرباندهای پایینی، کوچک و یا صفر خواهند بود (به دلیل ویژگی وجود شباهت در بین زیرباندهای هم‌راستا در تجزیه‌ی هرمی موجک)؛ حال با توجه به شباهت زیرباندهای هم‌راستا در تبدیل موجک

می‌توان انتظار داشت که استفاده از تکنیک انطباق بلوک در افزایش تعداد چنین مجموعه‌های ضرایب کوچک و یا صفر اثر قابل توجهی داشته باشد.

ویژگی دیگر عمده‌ی کدگذارهای SPC این است که هرچه ضرایب مهم و بزرگ خوشه‌ی بزرگتری تشکیل دهند، کارایی کدگذاری چنین کدگذارهایی نیز افزایش می‌یابد. تکنیک پیش پردازشی پیشنهادی در این پایان نامه که مبتنی بر دوباره‌چینی بلوکها و قرار دادن آنها برطبق متوسط اندازه‌ی ضرایب است، در جهت استفاده از این ویژگی می‌باشد. این تکنیک از دو نظر موجب افزایش کارایی فشرده‌سازی روش پیشنهادی می‌شود: اول، در هنگام پردازش فریمهای مستقل، تکنیک پیش‌پردازشی مذکور در حوزه‌ی تبدیل DCT استفاده شده و موجب قرارگرفتن مولفه‌های فرکانسی مهم و غالب در اوایل طیف می‌شود؛ بنابراین، تصویری پایین‌گذر تولید شده که مولفه‌های فرکانس بالای ناچیز و یا درحد صفر خواهد داشت. بنابراین، در زیرباندهای حاصل از تجزیه به کمک تبدیل موجک گروههای بزرگ و زیادی از ضرایب صفر یا نزدیک به آن به وجود خواهند آمد که کمک زیادی به افزایش کارایی فشرده‌سازی خواهند کرد. دوم، در هنگام پردازش فریمهای وابسته، پس از گرفتن تبدیل موجک، تکنیک پیش‌پردازشی مورد نظر مجدداً استفاده می‌شود که باعث در کنار هم قرار دادن ضرایب مهم (و تشکیل گروههای نسبتاً بزرگ از آنها) و نیز در کنار هم قرار دادن ضرایب کوچک (و در نتیجه، تشکیل گروههای نسبتاً بزرگ از آنها) خواهد شد. به این ترتیب، مجدداً کارایی کدگذارهای SPC افزایش خواهد یافت.

در این قسمت، در قالب تعدادی آزمایش، عملکرد و تاثیر هر یک از دو روش انطباق بلوک و بازچینی بلوکها به تنهایی و یا همزمان با هم روی کارایی کدگذاری دو کدگذار SPC شامل SPIHT و EBCOT بررسی می‌شود. معیارهای کمی که در این قسمت جهت ارزیابی نحوه‌ی عملکرد کدگذارهای SPC (شامل SPIHT و EBCOT) استفاده می‌شوند شامل متوسط PSNR (برحسب dB) و واریانس درصدی (نسبت به مقدار متوسط) در نرخ بیت ثابت (برحسب bpp) و از قبل مشخص شده است. در این خصوص به چند نکته باید توجه شود: اول این که منظور از متوسط PSNR و واریانس درصدی این است که مقدار PSNR برای هر فریم بازسازی شده (ورزش فوتبال) محاسبه و سپس متوسط و واریانس مقادیر به دست آمده محاسبه و منظور می‌شود. واریانس درصدی

نسبت به متوسط مذکور محاسبه می‌شود. برای مثال اگر متوسط و واریانس PSNR برابر به ترتیب ۴۰ و ۱۰ محاسبه شوند، واریانس درصدی به صورت ۲۵٪ گزارش می‌شود. دوم این که در این بخش، از آنجا که تمرکز اصلی آزمایشها هربار روی یک تصویر یا فریم است، از معیار نرخ بیت یا بیت بر پیکسل (برحسب bpp) به عنوان معیاری برای میزان فشردگی استفاده می‌شود اما در بخش بعدی که دنباله‌های ویدیویی به عنوان مجموعه‌ای از فریمها در محور زمان مورد بررسی می‌شوند از معیار نرخ ارسال بیت یا بیت بر ثانیه (برحسب bps) به عنوان معیاری برای میزان فشردگی استفاده می‌شود.

آزمایشهای انجام شده در این بخش به چهار دسته تقسیم شده‌اند که همگی آنها در نرخ بیت ۰/۱ bpp انجام می‌شوند: در دسته اول، از هر تصویر تبدیل موجک گرفته و از یکی از کدگذارهای SPIHT و یا EBCOT برای کدگذاری استفاده می‌کنیم. سپس میانگین و واریانس مقادیر PSNR (به دست آمده برای هر تصویر) را در نرخ بیت ثابت ۰/۱ bpp گزارش می‌دهیم. در دسته دوم، ابتدا از تصویر تبدیل موجک گرفته و تکنیک انطباق بلوکی را در حوزه تبدیل اعمال کرده و سپس از یکی از کدگذارهای SPIHT و یا EBCOT برای کدگذاری استفاده می‌کنیم. سپس میانگین و واریانس مقادیر PSNR (به دست آمده برای هر تصویر) را در نرخ بیت ثابت ۰/۱ bpp گزارش می‌دهیم. در دسته سوم مشابه با دسته دوم عمل می‌کنیم اما به جای انطباق بلوکی، از روش بازچینی بلوکها استفاده می‌کنیم. بالاخره در دسته چهارم نیز، ابتدا از تصویر تبدیل موجک گرفته و به ترتیب، تکنیکهای انطباق بلوکی و بازچینی بلوکی را در حوزه تبدیل اعمال کرده و سپس از یکی از کدگذارهای SPIHT و یا EBCOT برای کدگذاری استفاده می‌کنیم. سپس میانگین و واریانس مقادیر PSNR (به دست آمده برای هر تصویر) را در نرخ بیت ثابت ۰/۱ bpp گزارش می‌دهیم. در دسته چهارم توجه شود که ابتدا تکنیک انطباق بلوکی و سپس بازچینی بلوکی استفاده می‌شود نه عکس آن؛ زیرا در ابتدا که (پس از تبدیل موجک) شباهت بین زیرباندها وجود دارد، تکنیک انطباق بلوکی سعی در کاهش چنین تزیادهایی و در نتیجه، افزایش بلوکهای ضرایب کوچک و یا صفر دارد. حال در ادامه، با کمک تکنیک بازچینی بلوکی، بلوکهای هم‌مقدار را در

کنار هم قرار می‌دهیم تا کارایی کدگذار افزایش یابد. نتایج هر کدام از چهار دسته‌ی فوق به ترتیب در جدولهای ۱-۶ تا ۴-۶ نشان داده شده است.

جدول ۱-۶: میانگین PSNR (برحسب dB) و واریانس آن (برحسب درصد نسبت به میانگین) در نرخ بیت ۰/۱ bpp برای تصاویر استفاده شده و دسته‌ی اول آزمایشها (بدون اعمال هر کدام از دو بهبود پیشنهادی).

EBCOT	SPIHT	
۳۸/۸	۳۸/۱	میانگین
۱۲/۹	۱۴/۷	واریانس (درصد نسبت به میانگین)

جدول ۲-۶: میانگین PSNR (برحسب dB) و واریانس آن (برحسب درصد نسبت به میانگین) در نرخ بیت ۰/۱ bpp برای تصاویر استفاده شده و دسته‌ی دوم آزمایشها (اعمال تکنیک انطباق بلوکی پس از تبدیل موجک).

EBCOT	SPIHT	
۴۰/۶	۳۹/۳	میانگین
۱۴/۴	۱۷/۲	واریانس (درصد نسبت به میانگین)

جدول ۳-۶: میانگین PSNR (برحسب dB) و واریانس آن (برحسب درصد نسبت به میانگین) در نرخ بیت ۰/۱ bpp برای تصاویر استفاده شده و دسته‌ی سوم آزمایشها (اعمال تکنیک بازچینی بلوکها).

EBCOT	SPIHT	
۴۰/۲	۳۹/۶	میانگین
۱۲/۳	۱۱/۳	واریانس (درصد نسبت به میانگین)

جدول ۴-۶: میانگین PSNR (برحسب dB) و واریانس آن (برحسب درصد نسبت به میانگین) در نرخ بیت ۰/۱bpp برای تصاویر استفاده شده و دسته‌ی چهارم آزمایشها (اعمال هر دو بهبود پیشنهادی).

EBCOT	SPIHT	
۴۰/۹	۴۰	میانگین
۱۳/۱	۱۵/۲	واریانس (درصد نسبت به میانگین)

با توجه به نتایج جدول ۲-۶ تکنیک انطباق بلوکی، تاثیر مثبت بیشتری (نسبت به کدگذار SPIHT) روی کدگذار EBCOT داشته است. علت آن به نظر این می‌رسد که این کدگذار، برخلاف SPIHT کار خود را به صورت بلوک به بلوک انجام می‌دهد اما تکنیک SPIHT کار خود را به صورت ریشه به ریشه (یا باند به باند) انجام می‌دهد. همچنین استفاده از این تکنیک موجب افزایش واریانس PSNR شده است که مقدار آن در مورد کدگذار SPIHT بیشتر از کدگذار EBCOT است. افزایش واریانس به معنای این است که تکنیک پیشنهادی در مورد تصاویر مختلف بسیار متنوع عمل کرده است. در حالت کلی می‌توان گفت انطباق بلوکی تاثیر مثبت‌تری روی کدگذار EBCOT دارد.

با توجه به نتایج جدول ۳-۶ ملاحظه می‌شود که تکنیک بازچینی بلوکی تاثیر مثبت تقریباً یکسانی روی هر دو کدگذار SPIHT و EBCOT داشته است. البته، برخلاف حالت قبلی، در مورد هر دو کدگذار، واریانس مقادیر PSNR نسبت به حالت اولیه (یعنی بدون استفاده از هیچ روش بهبودی) کمتر شده است. این امر نشان دهنده‌ی این مطلب است که تکنیک بازچینی بلوکها در مورد همه‌ی انواع تصاویر ورزشی تاثیر مثبت و نسبتاً قابل قبولی داشته است.

هنگامی که هر دو روش بهبود پیشنهادی را به ترتیبی که بیان شد استفاده کنیم مقدار بهبود نهایی در PSNR در مورد هر روش SPIHT و EBCOT بیشتر از مقدار متناظر در مورد هر کدام از آزمایشهای دسته‌ی دوم و سوم به تنهایی است (جدول ۴-۶). اما این مقدار کمتر از مجموع مقادیر متناظر مذکور است؛ به بیان دیگر هنگامی که دو بهبود پیشنهادی را همزمان به تصاویر اعمال می‌کنیم، مقدار افزایش PSNR (در همان نرخ بیت bps) کمتر از مجموع مقادیر افزایش PSNR در هر کدام از آزمایشهای دسته‌ی دوم و سوم است. علت این امر به نظر می‌رسد این باشد که در اثر انجام انطباق بلوکی، واریانس بلوکهای باقیمانده بیشتر از واریانس بلوکهای اولیه شده و در نتیجه کارایی کدگذارهای SPC را که وابسته به منسجم و یکدست بودن ضرایب (خیلی کوچک و خیلی بزرگ) دارند، کاهش می‌دهد.

نتیجه‌ای که از این قسمت و با توجه به جدولهای ۱-۶ تا ۴-۶ می‌توان گرفت اثر مثبت نسبتاً قابل توجه دو روش بهبود پیشنهادی در افزایش کارایی کدگذارهای SPC است.

۳-۶ بررسی کارایی فشرده‌سازی روش پیشنهادی

در این قسمت، کارایی فشرده‌سازی روش پیشنهادی که روندنمای آن در شکل ۲-۵ نمایش داده شد، بررسی می‌گردد. در این بررسی، حالت‌های کاری اختیاری (زیربخش ۵-۳-۵) در نظر گرفته نمی‌شود. اثر این حالت‌های کاری اختیاری در بخش بعدی بررسی خواهد شد. معیارهای کمی که در این بخش از آنها جهت ارزیابی کارایی

فشرده‌سازی روش پیشنهادی و دیگر روشهای مرجع استفاده شده‌اند، شامل معیار متوسط PSNR (برحسب dB) و واریانس درصدی (نسبت به مقدار متوسط) در نرخ ارسال بیت (برحسب bps) مشخص می‌باشند. منظور از متوسط PSNR این است که مقدار PSNR برای تمام فریمهای مورد بررسی محاسبه و سپس متوسط (و البته) واریانس آنها محاسبه و منظور می‌شود.

روش پیشنهادی در بستر نرم‌افزار MATLAB روی رایانه‌ای با حافظه‌ی RAM مقدار ۴ گیگابایت و پردازنده‌ی اینتل از نوع Core i5 پیاده‌سازی گردید. روش پیشنهادی روی تعدادی دنباله‌های ویدیویی ورزش فوتبال که مشخصات آن در ابتدای فصل قبل ذکر گردید، اجرا گردید. با تنظیم روش پیشنهادی در نرخ ارسال بیت تقریبی ۳۲Kbps (سرعتی در کاربردهای ارسال با نرخ بیت بسیار پایین^{۱۸۰})، متوسط (برحسب dB) و واریانس درصدی مقدار PSNR روش پیشنهادی با مقدار مربوط به روشهای MPEG4 و FLV در جدول ۶-۵ مقایسه شده است. از آنجا که بستر نرم‌افزاری پیاده‌سازی سه روش مذکور مختلف بود، مقایسه مدت زمان اجرای سه روش مذکور منطقی به نظر نمی‌رسید. برای پیاده‌سازی دو روش MPEG4 و FLV از نرم‌افزار AVS Video Converter و برای پیاده‌سازی روش پیشنهادی از نرم‌افزار MATLAB استفاده گردید. روش فشرده‌سازی MPEG4 یکی از مهمترین استانداردهای فشرده‌سازی امروزی جهت ذخیره و ارسال اطلاعات ویدیویی در بستر شبکه اینترنت است که جزئیاتی از آن در فصل دوم بیان گردید. روش FLV نیز یکی از روشهای غیراستاندارد موجود امروزی جهت انتقال فایل‌های ویدیویی در بستر شبکه‌ی اینترنت است که نسبتاً متداول است.

¹⁸⁰Very Low Bit Rate Transmission

جدول ۶-۵: مقدار متوسط PSNR (برحسب dB) و واریانس (برحسب درصدی از مقدار متوسط) روش پیشنهادی و دو

روش MPEG4 و FLV در نرخ بیت ارسال ثابت ۳۲Kbps

FLV	MPEG4	Proposed (EBCOT Coder)	Proposed (SPIHT Coder)	
۳۳/۵	۳۴/۸	۳۶/۹	۳۶/۲	میانگین
۸/۴	۱۰/۵	۱۵/۸	۱۷/۳	واریانس (درصد) نسبت به میانگین

۶-۴ بررسی میزان تاثیر حالت‌های کاری اختیاری

همان طور که در فصل قبل بیان شد، روش پیشنهادی برای رسیدن به نرخ بیت‌های پایین‌تر دارای سه حالت کاری دیگر نیز می‌باشد که هر کدام به تنهایی یا با هم قابل استفاده‌اند. حالت کاری اول شامل کاهش نرخ نمونه برداری زمانی و به عبارت دیگر کاهش تعداد فریم‌ها در واحد زمان است. در این حالت از هر دو فریم، اطلاعات یکی حذف شده و کدگذاری نمی‌شود. در موقع بازسازی، برای رسیدن به نرخ فریم اولیه، از هر دو فریم متوالی یک فریم را به روش درونیابی تولید و بین آن دو فریم مربوطه جایگذاری می‌کنیم تا به همان نرخ فریم اولیه برسیم.

در حالت کاری دوم، نرخ نمونه‌برداری مکانی هر فریم را کاهش می‌دهیم. برای این کار، در هر سطر و هر ستون از یک فریم، از هر دو پیکسل یکی را حذف می‌کنیم. به عبارت دیگر در هر بلوک 2×2 از پیکسل‌ها دو پیکسل مشخص را نگاه داشته و مابقی را حذف می‌کنیم. به این ترتیب ابعاد هر فریم به نصف ابعاد اولیه تقلیل می‌یابد. در هنگام بازسازی، برای رسیدن به همان نرخ نمونه‌برداری مکانی اولیه، بین هر دو پیکسل مجاور، یک پیکسل جدید از طریق درونیابی محاسبه و جایگذاری می‌کنیم.

حالت کاری سوم روش پیشنهادی، کاهش دقت بیتی استفاده شده برای نمایش مقادیر مولفه‌های رنگ پیکسلها است. برای نمایش هر مولفه‌ی رنگ از یک پیکسل، معمولاً از ۸ بیت استفاده می‌شود. در این حالت کاری، از روش کوانتیزاسیون ساده (یکنواخت) برای کاهش تعداد رنگ هر پیکسل استفاده می‌کنیم تا دقت بیتی برای نمایش هر مولفه‌ی رنگ از ۸ بیت به ۶ بیت کاهش یابد. برای انجام این نوع کوانتیزاسیون، مقدار هر پیکسل بر ۴ تقسیم شده و نتیجه به نزدیکترین عدد صحیح گرد می‌شود. می‌دانیم که تقسیم بر ۴ معادل با از دست دادن ۲ بیت دقت نمایش است. در هنگام بازسازی، مقادیر ۶ بیتی بازسازی شده در ۴ ضرب می‌شوند تا تقریبی از دقت ۸ بیتی اولیه به دست آید.

در این قسمت، روش پیشنهادی (دیگرام بلوکی شکل ۵-۲) را با یکی یا تمام سه حالت کاری مذکور ترکیب کرده و مورد ارزیابی قرار می‌دهیم. بنابراین، چهار حالت مختلف پیش می‌آید که در هر حالت، میزان کاهش درصدی متوسط نرخ بیت را نسبت به حالت اصلی (یعنی بدون استفاده از حالت‌های کاری اختیاری پیشنهادی) گزارش می‌دهیم. این نتایج در جدول ۶-۶ آورده شده‌اند. در استفاده از این جدول توجه کنید که دیگر شرط ثابت بودن نرخ ارسال را در نظر نگرفته‌ایم، بنابراین، در استفاده از حالت‌های کاری، ممکن است مقدار نرخ ارسال کمتر یا بیشتر از ۳۲ Kbps گردد. در حقیقت هرگاه حالت‌های کاری منجر به فشرده‌سازی بیشتر شوند، حجم بیت‌های لازم برای ارسال کمتر شده و با نرخ ارسال کمتری می‌توان آنها را ارسال نمود؛ اما در مقابل، هرگاه بسته به شرایط فریمها، کارایی فشرده‌سازی حالت‌های کاری پیشنهادی افت پیدا کند، حجم بیت‌های لازم جهت ارسال، افزایش یافته و نیاز به نرخ ارسال بیشتری خواهیم داشت. بنابراین، مقایسه روش پیشنهادی در این حالت با روشهای MPEG4 و FLV چندان جایگاهی نخواهد داشت. همچنین توجه کنید که در جدول ۶-۶ از معیار نرخ بیت با واحد بیت بر پیکسل (bpp) (و نه معیار PSNR) جهت سنجش کارایی فشرده‌سازی استفاده شده است.

هرقدر این مقدار کمتر شود میزان فشردگی^{۱۸۱} بیشتر خواهد بود بنابراین، مقدار کاهش درصدی نرخ بیت معادل با مقدار افزایش درصدی میزان فشردگی است.

جدول ۶-۶: مقدار کاهش درصدی نرخ بیت در نتیجه استفاده از یکی یا تمام حالت‌های کاری پیشنهادی نسبت به حالت اصلی (یعنی عدم استفاده از هرگونه حالت کاری)

استفاده از حالت کاری اول	استفاده از حالت کاری دوم	استفاده از حالت کاری سوم	استفاده از تمام حالت‌های کاری
۱۵/۳٪	۱۲/۳٪	۸/۶٪	۲۷/۶٪
میانگین افزایش درصدی PSNR			

در حالت کاری اول، نیمی از داده‌های ویدیویی کاهش می‌یابد. البته در مقابل، تزیاید زمانی بین فریمها اندکی کاهش یافته و اثر انطباق بلوکی در محور زمان کاهش می‌یابد. در حالت کاری دوم، داده‌های ویدیویی تا حد یک دوم کاهش می‌یابد اما در مقابل، تزیاید مکانی و حتی تا حدی تزیاید زمانی (وقتی از هر بلوک نواحی ثابت و مشخصی حذف شوند، قسمتهایی از یک شیء موجود در فریمهای ویدیویی در برخی فریمها ظاهر و در برخی فریمها ناپدید می‌شود) تا حد نسبتاً قابل توجهی کاهش می‌یابند؛ بنابراین، کارایی دو نوع انطباق بلوکی مورد استفاده کاهش می‌یابد. در نتیجه، می‌توان انتظار داشت، حالت کاری دوم، اثر مثبت کمتری روی کاهش درصدی نرخ بیت (یا افزایش درصدی نرخ فشردگی) داشته باشد.

در حالت کاری سوم، تعداد کل بیتها و در نتیجه، حجم داده‌های ویدیویی کاهش می‌یابد اما این مقدار کاهش مطمئناً تا حد قابل توجهی، کمتر از مقدار کاهش حجم داده‌های ویدیویی در حالت‌های کاری اول و دوم است: در این حالت، در خوش‌بینانه‌ترین حالت، تعداد کل بیتها ۲۵٪ کاهش می‌یابد. البته در این حالت کاری، همچنان حداقل یکی از تزیاید‌های مکانی و زمانی در مقایسه با حالت‌های کاری اول و دوم، بیشتر بوده و در نتیجه در حالت

¹⁸¹ Compression Rate

کلی، انطباق بلوکی از کارایی بیشتری برخوردار خواهد بود. با توجه به دلایل فوق‌الذکر، نتایج جدول ۶-۶ چندان دور از انتظار نخواهد بود.

۶-۵ نتیجه‌گیری

نتایج عددی روش فشرده‌سازی پیشنهادی جهت دنباله‌های ویدیویی ورزش فوتبال در این فصل ارائه گردید. ابتدا کارایی و اثر دو روش بهبود کدگذاری پیشنهادی یعنی انطباق بلوکی و بازچینی بلوکی بررسی گردید. کدگذار EBCOT نسبت به کدگذار SPIHT سود بیشتری از روش انطباق بلوکی پیشنهادی برده است. دو روش بهبود پیشنهادی اثر مثبت تقریباً یکسانی روی دو کدگذار مزبور داشته‌اند. در ادامه، کارایی روش فشرده‌سازی پیشنهادی در دو حالت استفاده از کدگذار EBCOT و استفاده از کدگذار SPIHT با معیار متوسط PSNR (در نرخ ارسال ثابت) ارزیابی و نتایج مربوطه با نتایج مربوط به دو روش MPEG4 و FLV مقایسه گردیدند. برطبق این نتایج، روش پیشنهادی تا حدود حداکثر ۱۰ درصد، مقادیر متوسط PSNR بیشتری (نسبت به دو روش مذکور) تولید کرده است. در انتها، کارایی روش پیشنهادی در حالت استفاده از یکی یا تمام سه حالت کاری اختیاری پیشنهادی بررسی گردیده و نتایج آن با نتایج مربوط به حالت عدم استفاده از این حالت‌های کاری مقایسه گردید. در این مقایسه، نرخ ارسال دیگر ثابت در نظر گرفته نشده و از معیار نرخ بیت (که متناسب معکوس با میزان فشرده‌سازی است) استفاده گردید.

فصل هفتم

نتیجه گیری و پیشنهاد راهکارهای آینده

۱-۷ نتیجه گیری

در این پایان نامه روشی برای فشرده سازی دنباله های ویدیویی ورزش فوتبال پیشنهاد شد که براساس تعیین نوع فریمها، استفاده از تبدیل موجک و کدگذاری ضرایب آن به کمک برخی کدگذارهای SPC شامل SPIHT و EBCOT، و استفاده از برخی تکنیکها مانند انطباق بلوکی (در حوزه های مختلف) و بازچینی بلوکها استوار بود. همچنین در روش پیشنهادی، برخی حالت های کاری اختیاری شامل کاهش نرخ نمونه برداری زمانی، کاهش نرخ نمونه برداری مکانی، و کاهش نرخ نمونه برداری مقدار نیز معرفی گردید.

در روش پیشنهادی، به موازات ورود هر فریم ابتدا نوع آن از نظر مستقل یا وابسته بودن تعیین می شود. فریم مستقل، فریمی است که جداگانه کدگذاری می شود و فریم وابسته، فریمی است که با توجه به فریم کدگذاری شده ی قبلی کدگذاری می شود. هدف از استفاده از فریمهای وابسته افزایش میزان فشرده سازی (یا همان کارایی کدگذاری) است. معمولاً فریمهای هر نما همبستگی بسیار زیادی با یکدیگر دارند لذا استفاده از این وابستگی ها در جهت افزایش کارایی کدگذاری تاثیر قابل توجهی دارد. برحسب این که فریم جاری از نوع مستقل است یا وابسته یکی از دو مسیر مشخص شده از قبل طی می شود.

در مسیر اول فرض بر این است که فریم مستقل است. ابتدا از یک پیش پردازش مبتنی بر تبدیل DCT به منظور مرتب کردن مولفه های فرکانسی و در تجمیع و انباشتن مولفه های هم مقدار در کنار یکدیگر استفاده می شود. این کار باعث افزایش کارایی کدگذارهای SPC (مانند SPIHT و EBCOT) می گردد. پس از انجام پیش پردازش، از تصویر خروجی تبدیل موجک گرفته می شود. سپس، به منظور افزایش کارایی کدگذاری نهایی، از

انطباق بلوکی به منظور کاهش تزیاید (یا همبستگی) بین ضرایب موجک استفاده می‌شود. حال، در انتهای مسیر اول و پس از اجرای تکنیک انطباق بلوکی در حوزه‌ی ضرایب تبدیل موجک، از یکی از کدگذارهای SPC استفاده می‌شود تا دنباله‌ی بیتی نهایی تشکیل شود.

در مسیر دوم فرض بر این است که فریم فعلی از نوع وابسته است. بنابراین دارای شباهت (یا تزیاید) قابل توجهی با آخرین فریم کدگذاری شده است. ابتدا با استفاده از تکنیک انطباق بلوکی در حوزه‌ی مکان (یا پیکسلی) تزیاید مزبور تا حد امکان کاهش داده می‌شود. سپس از تصویر حاصل، تبدیل موجک گرفته شده و پیش پردازش بازچینی بلوکی روی تصویر (یا ماتریس) حاصل از تبدیل موجک اعمال می‌شود. در انتهای مسیر دوم از یکی از کدگذارهای SPC به منظور تولید دنباله‌ی بیتی خروجی استفاده می‌شود.

روش پیشنهادی برای رسیدن به نرخ بیت‌های پایین‌تر (و در نتیجه میزان فشرده‌سازی بیشتر) دارای سه حالت کاری دیگر نیز بوده است که هرکدام به تنهایی یا با هم قابل استفاده‌اند. حالت کاری اول شامل کاهش نرخ نمونه‌برداری زمانی و به عبارت دیگر کاهش تعداد فریمها در واحد زمان است. ویدیوها معمولاً با نرخ ۲۵ الی ۳۰ فریم بر ثانیه تولید می‌شوند. در این حالت از هر دو فریم، اطلاعات یکی حذف شده و کدگذاری نمی‌شود. با این کار نرخ فریمها به نصف تقلیل می‌یابد. در موقع بازسازی، برای رسیدن به نرخ فریم اولیه، از هر دو فریم متوالی یک فریم به روش درونیایی تولید و بین آن دو فریم مربوطه جایگذاری گردید تا نرخ فریم اولیه بازسازی گردد. در حالت کاری دوم، نرخ نمونه‌برداری مکانی هر فریم کاهش داده شد. برای این کار، در هر سطر و هر ستون از یک فریم، از هر دو پیکسل یکی حذف می‌شود. به این ترتیب ابعاد هر فریم به یک دوم اولیه تقلیل می‌یابد. در هنگام بازسازی، برای رسیدن به همان نرخ نمونه‌برداری مکانی اولیه، بین هر دو پیکسل مجاور، یک پیکسل جدید از طریق درونیایی محاسبه و جایگذاری می‌گردد.

حالت کاری سوم روش پیشنهادی، کاهش دقت بیتی استفاده شده برای نمایش مقادیر مولفه‌های رنگ پیکسلها بوده است. برای نمایش هر پیکسل، معمولاً از ۸ بیت برای نمایش هر یک از مولفه‌های رنگ استفاده می‌شود. در

حالت کاری سوم، از روش کوانتیزاسیون ساده (یکنواخت) برای کاهش تعداد رنگ هر پیکسل استفاده می‌شود تا دقت بیتی برای نمایش هر مولفه‌ی رنگ از ۸ بیت به ۶ بیت کاهش یابد.

روش پیشنهادی قادر به تشخیص نواحی مطلوب در فریمهای ورزش فوتبال براساس ویژگیهایی مانند رنگ می‌باشد. پس از جداسازی نواحی مطلوب از دیگر نواحی، این اطلاعات در کدگذارهای SPC جهت کدگذاری خاص استفاده می‌شود. در حقیقت، برای نواحی مطلوب، نرخ بیت بیشتری تخصیص داده می‌شود.

در هنگام ارزیابی کارایی روش پیشنهادی، ابتدا کارایی و اثر دو روش بهبود کدگذاری پیشنهادی یعنی انطباق بلوکی و بازچینی بلوکی بررسی گردید. مشخص گردید که کدگذار EBCOT نسبت به کدگذار SPIHT سود بیشتری از روش انطباق بلوکی پیشنهادی می‌برد. دو روش بهبود پیشنهادی اثر مثبت تقریباً یکسانی روی دو کدگذار مزبور داشتند. در ادامه، کارایی روش فشرده‌سازی پیشنهادی در دو حالت استفاده از کدگذار EBCOT و استفاده از کدگذار SPIHT با معیار متوسط PSNR (در نرخ ارسال ثابت) ارزیابی و نتایج مربوطه با نتایج مربوط به دو روش MPEG4 و FLV مقایسه گردیدند. برطبق این نتایج، روش پیشنهادی تا حدود حداکثر ۱۰ درصد مقادیر متوسط PSNR بیشتری (نسبت به دو روش مذکور) تولید کرده است. در انتها، کارایی روش پیشنهادی در حالت استفاده از یکی یا تمام سه حالت کاری اختیاری پیشنهادی بررسی گردیده و نتایج آن با نتایج مربوط به حالت عدم استفاده از این حالت‌های کاری مقایسه گردید. در این مقایسه، نرخ ارسال دیگر ثابت در نظر گرفته نشده و از معیار نرخ بیت (که متناسب معکوس با میزان فشرده‌سازی است) استفاده گردید.

۲-۷ پیشنهاد راهکارهای آینده

به عنوان یکی از راهکارهای افزایش کارایی روش پیشنهادی، توصیه می‌گردد از روشهای ناحیه‌بندی دقیقتر و پیشرفته‌تری برای جداسازی و تعیین نواحی مطلوب استفاده شود. همچنین شاید لازم باشد تعریف نواحی مطلوب مورد بازبینی و تجدیدنظر قرار بگیرد. ممکن است بهتر باشد به جای تعریف دو ناحیه (مطلوب و عادی) از چند ناحیه با درجه‌های اهمیت مختلف استفاده شود. پیشنهاد بعدی ما در این پایان نامه، استفاده از دیگر روشهای کدگذاری SPC مانند WBTC و ارزیابی کارایی فشرده‌سازی آنها در کاربرد فشرده‌سازی مبتنی بر ناحیه‌ی مطلوب است. به عنوان آخرین پیشنهاد، بررسی حجم محاسباتی روش پیشنهادی و بررسی قابلیت پیاده‌سازی بلادرنگ آن می‌توان موضوع مناسبی برای ادامه‌ی کار باشد.

مراجع

- [1]. JAIN, A.K.: *Fundamentals of Digital Image Processing*, Prentice Hall, Upper Saddle River, 1989.
- [2]. CHEN, W., SMITH, C. and FRALICK, S.: ‘A fast computational algorithm for the discrete cosine transform’, *IEEE Trans. Commun.*, 1979, **COM-25**, pp. 1004–1009.
- [3]. ISHIGURO, T. and IINUMA, K.: ‘Television bandwidth compression transmission by motion-compensated interframe coding’, *IEEE Commun. Mag.*, 1982, **10**, pp. 24–30.
- [4]. KAPPAGANTULA, S. and RAO, K.R.: ‘Motion compensated predictive coding’, Proceedings of International Technical Symposium, SPIE, San Diego, CA, August 1983
- [5]. BERGMANN, H.C.: ‘Displacement estimation based on the correlation of image segments’, IRE Conference on the Electronic Image Processing, York, UK, July 1982
- [6]. JAIN, J.R. and JAIN, A.K.: ‘Displacement measurement and its application in interframe image coding’, *IEEE Trans. Commun.*, 1981, **COM-29**, pp. 1799–1808
- [7]. SHANABLEH, T. and GHANBARI, M.: ‘Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats’, *IEEE Trans. Multimedia*, 2002, **2:2**, pp. 101–110.
- [8]. KOGA, T., IINUMA, K., HIRANO, A., IJIMA, Y. and ISHIGURO, T.: ‘Motion compensated interframe coding for video conferencing’, Proceedings of National Telecommunication Conference, New Orleans, LA, 29 November– 3 December 1981, pp. G5.3.1–G5.3.5.
- [9]. GHANBARI, M.: ‘The cross search algorithm for motion estimation’, *IEEE Trans. Commun.*, 1990, **38:7**, pp. 950–953.
- [10]. PURI, A., HANG, H.M. and SCHILLING, D.L.: ‘An efficient blockmatching algorithm for motion compensated coding’, Proceedings of IEEE ICASSP’87, 1987, pp. 25.4.1–25.4.4.
- [11]. M. Ghanbari, “Standard Codecs, Image Compression to advanced video coding,” *IET Press*, 3rd Edition, 2011.

- [12]. PLOMPEN, R.H.J.M.: 'Motion video coding for visual telephony' (Proefschrift, 1989).
- [13]. NGAN, K.N.: 'Two-dimensional transform domain decimation technique', IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP'86, 1986, pp. 1001–1004.
- [14]. CHEN, C.T. and WONG, A.: 'A self-governing rate buffer control strategy for pseudoconstant bit rate video coding', *IEEE Trans. Image Process.*, 1993, **2:1**, pp. 50–59.
- [15]. CARR, M.D.: 'Video codec hardware to realise a new world standard', *Br. Telecom. J.*, 1990, **8:3**, pp. 28–35
- [16]. OKUBA, S., MCCANN, K. and LIPPMAN, A.: 'MPEG-2 requirements, profile and performance verification', *Signal Process., Image Commun.*, 1995, **7:3**, pp. 201–209.
- [17]. SAVATIER, T.: 'Difference between MPEG-1 and MPEG-2 video'. ISO/ IEC JTC1/SC29/WG11 MPEG94/37, March 1994.
- [18]. Draft ITU-T Recommendation H.263p: 'Video coding for very low bit rate communication', September 1997.
- [19]. ITU-T recommendation H.263p: 'Video coding for low bit rate communication', ITU-T SG16, February 2000.
- [20]. WIEGAND, T.: 'H.26L test model long-term number 9 (TML-9) draft0', VCEG-N83 d1, Germany, December 2001.
- [21]. KOENEN, R., PEREIRA, F. and CHIARIGLIONE, L.: 'MPEG-4: context and objectives', *Image Commun. J.*, 1997, **9:4**, pp. 295–304.
- [22]. MPEG-4: 'Generic coding of audio-visual objects: Part 2 - visual', ISO/IEC JTC1/SC29/WG11 N1902, FDIS of ISO/IEC 14496-2, Atlantic City, November 1998.
- [23]. MPEG-4 video verification model version-11, ISO/IEC JTC1/SC29/WG11, N2171, Tokyo, March 1998.
- [24]. WIEGAND, T., SULLIVAN, G.J., BJONTEGARD, G. and LUTHRA, A.: 'Overview of the H.264/AVC video coding standard', *IEEE Trans. Circuits Syst. Video Technol.*, 2003, **13:7**, pp. 560–576.

- [25]. ISO/IEC JTC1/SC29/WG11/N4031: 'Overview of the MPEG-7 standard', Singapore, March 2001.
- [26]. ISO/IEC JTC1/SC29/WG11/N4040: 'Study on MPEG-21 (digital audiovisual framework) Part 1', Singapore, March 2001.
- [27]. SHAPIRO, J.M.: 'Embedded image coding using zero trees of wavelet coefficients', *IEEE Trans. Signal Process.*, 1993, **4:12**, pp. 3445–3462.
- [28]. USEVITCH, B.E.: 'A tutorial on modern lossy wavelet image compression: foundation of JPEG 2000', *IEEE Signal Process. Mag.*, 2001, pp. 22–35.
- [29]. CROCHIERE, R.E., WEBER, S.A. and FLANAGAN, J.L.: 'Digital coding of speech in sub bands', *Bell Syst. Tech. J.*, 1967, **55**, pp. 1069–1085.
- [30]. DAUBECHIES, I.: 'Orthonormal bases of compactly supported wavelets', *Commun. Pure Appl. Math.*, 1988, **41**, pp. 909–996.
- [31]. SAID, A. and PEARLMAN, W.A.: 'A new, fast and efficient image codec based on set partitioning in hierarchical trees', *IEEE Trans. Circuits Syst. Video Technol.*, 1996, **6:3**, pp. 243–250.
- [32]. TAUBMAN, D., ORDENLICH, E., WEINBERGER, M. and SEROUSSI, G.: 'Embedded block coding in JPEG2000', *Signal Process. Image Commun.*, 2002, **17:1**, pp. 1–24.
- [33]. TAUBMAN, D.: 'High performance scalable image compression with EBCOT', *IEEE Trans. Image Process.*, 2000, **9:7**, pp. 1158–1170.
- [۳۴] هادی گرایلو، رویا سلطانی، و مصطفی صفایی، "ارائه روشی برای آشکارسازی و تعیین نوع تغییر نما در دنباله های ویدئویی و مناسب برای پیاده سازی سخت افزاری زمان حقیقی در بستر , FPGA و پردازنده های DSP" هفتمین کنفرانس ماشین بینایی و پردازش تصویر ایران، دانشگاه علم و صنعت ایران، آبان ۱۳۹۰.
- [35]. PURI, A., HANG, H.M. and SCHILLING, D.L.: 'An efficient blockmatching algorithm for motion compensated coding', *Proceedings of IEEE ICASSP'87*, 1987, pp. 25.4.1–25.4.4.

- [36]. BIERLING, M.: 'Displacement estimation by hierarchical block matching', *Proc. SPIE Vis. Commun. Image Process.*, 1988, **1001**, pp. 942–951.

Abstract

Today, many organizations are producing, storing, and transmitting a large amount of video data which, therefore, needs a large amount of memory; but, this is not commercially beneficial and also, takes large physical space. Therefore, it is difficult to displace and access such data storage devices. An obvious solution to this problem is to employ efficient video compression techniques. Region-Of-Interest (ROI) video coding is a newly introduced idea which has application in those video data in which all regions of a frame have not equal importance. In this technique, more bit budget is spent for more important regions and thus, they will have better quality than other regions. It seems that the ROI video coding idea has acceptable performance in medical, sport, and personal images.

In this thesis, a ROI soccer sport video compression method is proposed in which we define two different regions and encode them separately on the basis of their importance. The proposed method uses wavelet transform and encodes the wavelet coefficients using SPC coders especially SPIHT and EBCOT. Most important employed techniques include classifying frame into two types of independent and dependent ones using a shot boundary detection method, and encoding each type differently, using block matching technique in both spatial and transform domain, using block reordering technique, and finally, defining and determining interested regions in soccer video frames. The compression efficiency of the proposed compression method is compared to that of MPEG4 and FLV video formats using PSNR, bit per pixel (bpp). And bit per second (bps) measures. Simulation results show that the proposed methods outperforms, in average, to the mentioned ones.



Shahrood University of Technology

Faculty of Electrical Engineering

**Design and Implementation of a ROI CODEC for Sport
Videos**

**Thesis Submitted in partial Fulfillment of The Requirement for The Degree of
Master of Science**

(M.SC.)

RoyaSoltani

Supervisor

Dr.HadiGrailu

Associate Supervisor

Dr.AlirezaAhmadifard

Date: January 2013