

روشی جدید برای فشرده‌سازی ویدئو بدون نیاز به ارسال فریم تفاضل

فاطمه باقری

دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
fbagheri@ce.aut.ac.ir

شهره کسایی

دانشیار، دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف
skasaei@sharif.edu

مجید زیارت‌بان

دانشکده مهندسی برق
دانشگاه صنعتی امیرکبیر
m_ziaratban@aut.ac.ir

در روش پیشنهادی، جبران‌سازی بصورت منطقی‌تری انجام شده و با انجام جبران‌سازی در چند مرحله متوالی، فریم تفاضل دارای انتروپی بسیار کمی خواهد شد و بنابراین برای کدگشا فرستاده نمی‌شود. منطقی‌تر شدن عملیات، بدلیل انجام جبران‌سازی شدت روشنایی متوسط است که در روش پیشنهادی ارائه شده است. در بخش ۲ ابتدا روشهای مختلف تخمین حرکت معرفی شده و روشی که در این مقاله مورد استفاده قرار گرفته است، بیان می‌شود. در بخش ۳ توضیحات کاملی در مورد تغییرات اعمالی به الگوریتم پایه و ارائه روش جدید فشرده‌سازی ویدئو ذکر خواهد شد. نتایج آزمایشات در بخش ۴ آورده شده و بخش ۵ به جمع‌بندی و نتیجه‌گیری می‌پردازد.

۲- تخمین حرکت

روشهای بسیاری برای تخمین حرکت ارائه شده‌است. اولین و ساده‌ترین روش، روش جستجوی کامل است. این روش بدلیل جستجوی کاملی که در ناحیه جستجو انجام می‌دهد بهترین تخمین را ارائه می‌نماید، اما بار محاسباتی آن با افزایش محدوده جستجو، بشدت افزایش می‌یابد. به همین منظور روشهای دیگری پیشنهاد شدند که تعداد مقایسه‌ها را کاهش داده و سرعت را افزایش می‌دهند. تعدادی از این روشها عبارتند از روش سه مرحله‌ای [1]، تخمین مبتنی بر لبه [2]، الگوریتم لگاریتمی دوبعدی [3]، تخمین مبتنی بر گرادیان [4]، روش جستجوی یک‌بعدی هرمی موازی [5] و جستجوی هرمی [6-9]. در این مقاله، از الگوریتم لگاریتمی دوبعدی استفاده شده است که روند جستجوی آن در شکل ۱ نشان داده شده است.

چکیده: در فشرده‌سازی‌های پایه (MPEG 1,2)، پس از محاسبه بردارهای جابجایی، فریم تفاضل بین فریم جاری و فریم بازسازی شده قبلی نیز به‌همراه بردارهای جابجایی کد شده و برای کدگشا ارسال می‌شود. در این مقاله روشی پیشنهاد شده‌است که جبران‌سازی حرکت را به‌گونه‌ای انجام می‌دهد که نیازی به ارسال فریم تفاضل برای کدگشا نباشد. با تغییرات ایجاد شده در الگوریتم پایه، روش پیشنهادی سریعتر و منطقی‌تر عمل نموده و با حجم محاسبات کمتر، به تعداد بیت کمتری برای ارسال نیاز خواهد داشت.

کلمات کلیدی: فشرده‌سازی ویدئو، فریم تفاضل، جبران‌سازی حرکت، جبران‌سازی شدت روشنایی متوسط

۱- مقدمه

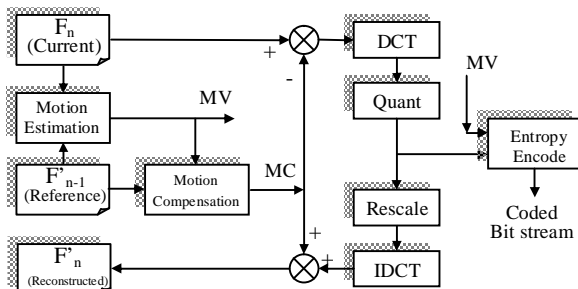
در فشرده‌سازی‌های پایه MPEG 1,2، به این صورت عمل می‌شود که با مقایسه فریم جاری و فریم بازسازی شده قبلی، تخمینی از میزان حرکت و جابجایی اشیای متحرک موجود در این دو فریم بدست می‌آید. سپس با جبران‌سازی آنها میزان تفاوت بین فریم جبران‌سازی شده و فریم جاری کاهش می‌یابد و بنابراین با تعداد بیت کمتری می‌توان فریم تفاضل بین این دو را برای کدگشا ارسال نمود. برای کد کردن این فریم تفاضل از تبدیل DCT استفاده می‌شود. پس از بدست آمدن ضرایب تبدیل DCT و کوانتیزه کردن آنها با ماتریس کوانتیزاسیون، برای فشرده‌سازی بیشتر از فشرده‌سازی‌های RLC و هافمن استفاده می‌شود. برای حفظ کیفیت ویدئو در بالاترین مقدار ممکن، باید مقدار درایه‌های ماتریس کوانتیزاسیون را با عدد یک مقداردهی نمود تا حداقل کاهش را در کیفیت داشته باشیم. اما از طرف دیگر اینکار باعث افزایش حجم اطلاعات خواهد شد. برای کاهش حجم داده‌های کد شده، از ماتریس کوانتیزاسیون با درایه‌های بزرگتر از یک استفاده می‌شود. هر قدر مقدار درایه‌ها بزرگتر انتخاب شوند، تعداد بیت کمتری مورد نیاز خواهد بود اما در عوض، کیفیت تصاویر بازسازی شده کاهش خواهد یافت.

۳-۲- استفاده از ساختار متوالی

همانطور که در قسمتهای قبل توضیح داده شد، در این روش فشرده‌سازی، هدف این است که فریم تفاضل ارسال نشود، تا بتوان تعداد بیت کمتری برای فشرده‌سازی بدست آورد. بنابراین نمی‌توان تخمین حرکت را با بلاکهای ۱۶×۱۶ انجام داد. هرچند با جبران‌سازی توسط این بلاکهای بزرگ نیز انتروپی فریم تفاضل و در نتیجه حداقل تعداد بیت مورد نیاز برای کد و ارسال آن کاهش می‌یابد، اما کیفیت فریم جبران‌سازی شده در حدی نیست که بتوان از آن بعنوان یک فریم مستقل در کدگشا استفاده کرد. برای حل این مشکل دو روش وجود دارد. اولین و ساده‌ترین روش اینست که برای نزدیکتر شدن فریم جبران‌سازی شده به فریم جاری باید اندازه بلاک را کمترین مقدار انتخاب نمود. بنابراین باید تخمین حرکت را با بلاکهای ۱×۱ یا همان پیکسلها انجام داد. اما مشکلی که این روش دارد آن است که حجم محاسباتی آن بسیار بالا است. دلیل این امر این است که برای بلاک ۱×۱ باید به تعداد پیکسلهای متحرک در دو فریم، عملیات تخمین حرکت را اجرا نمود که باعث کند شدن الگوریتم خواهد شد.

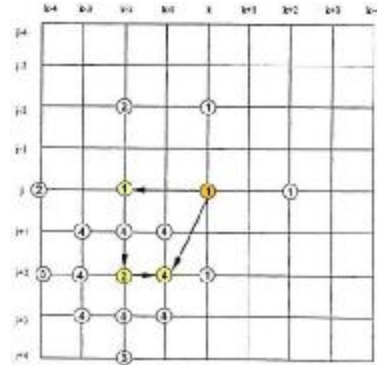
روش دیگر استفاده از ساختار متوالی است. شکل ۲ دیگرام کلی فشرده‌سازی پایه را نشان می‌دهد. همانطور که ملاحظه می‌شود با بدست آوردن فریم تفاضل و تخمین حرکت، بردارهای حرکت بدست آمده و برای کدگشا ارسال می‌شوند. فریم جبران‌سازی شده MC نیز از فریم جاری کم می‌شود و فریم تفاضل آنها پس از تبدیل DCT و فشرده‌سازی ضرایب بدست آمده، کد و ارسال می‌شوند.

برای نزدیکتر شدن کیفیت فریم جبران‌سازی شده به فریم جاری، می‌توان از متوالی کردن دیگرام شکل ۲، با اندازه بلاکهای مختلف استفاده کرد. این ساختار جدید در شکل ۳ نشان داده شده است. با پیشروی در دیگرام، در هر مرحله، اندازه بلاکها کوچکتر شده و در نهایت به مقدار ۱×۱ خواهد رسید.



شکل ۲: دیگرام کلی فشرده‌سازی MPEG

با این روش می‌توان تعداد عملیات لازم برای جبران‌سازی را تا حدود زیادی کاهش داد. با بکارگیری روش هرمی که در درون این ساختار متوالی نهفته است برای اندازه بلاکها، ابتدا جابجاییهای بزرگتر با بلاکهای بزرگتر تخمین زده شده و جبران می‌شوند. سپس در مرحله i ,



شکل ۱: الگوریتم جستجوی لگاریتمی دوبعدی

در این روش، جستجو در چند مرحله انجام می‌شود. در مرحله اول، ۵ ناحیه در همسایگی پیکسل مرجع که اندازه d از یکدیگر فاصله دارند در نظر گرفته شده و پس از مقایسه، بهترین آنها انتخاب می‌شود.

$$d = \lceil \log_2(p) \rceil + 1 \quad (1)$$

که p طول ناحیه جستجو را مشخص می‌کند. با بدست آمدن پیکسل جدید، جستجو در ۵ همسایگی آن با همان فاصله d انجام می‌شود. اینکار تا جایی ادامه می‌یابد که یکی از دو شرط زیر برآورده شوند.

- پیکسل برنده جدید همان پیکسل مرحله قبل باشد.
- پیکسل برنده بر روی مرز ناحیه جستجو واقع شود.

در اینصورت مقدار d نصف شده و مراحل فوق تکرار می‌شود. این روند تا جایی ادامه می‌یابد که مقدار d برابر با یک شود. در این مرحله بجای ۵ همسایگی، همه ۹ همسایگی بررسی خواهند شد. به این ترتیب مناسبترین بلاک توسط این روش تخمین زده خواهد شد.

۳-۳- روش جدید فشرده‌سازی ویدئو

در این مقاله روشی برای فشرده‌سازی ویدئو ارائه می‌شود که در آن نیازی به فشرده‌سازی و ارسال فریم تفاضل وجود نخواهد داشت. در این روش، سه تغییر کلی در الگوریتم پایه ایجاد شده است تا بتوان از ارسال فریم تفاضل صرفنظر کرد. این سه تغییر عبارتند از:

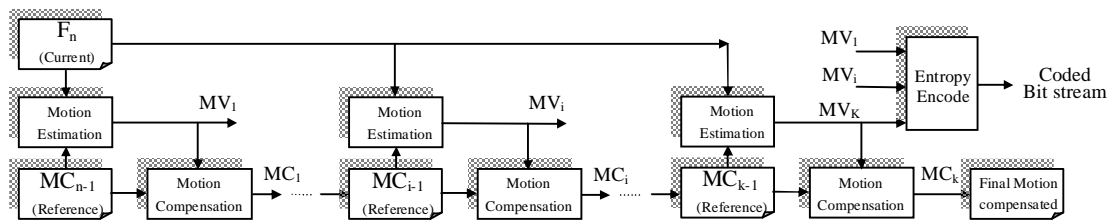
۳-۱- کاهش تعداد اجرای عملیات تخمین حرکت

اولین تغییری که در الگوریتم پایه ایجاد شده است، از این مفهوم استفاده می‌کند که جابجاییهای بسیار کم در زمینه یا بافتهای ساده در بین دو فریم متوالی برای انسان قابل توجه نبوده و نیازی به تخمین و جبران آنها نخواهیم داشت. بنابراین با استفاده از فریم تفاضل، تنها در قسمتهایی که قدر مطلق اختلاف دو فریم از حدی بیشتر باشد، میزان حرکت تخمین زده شده و جبران خواهد شد. با انجام این تغییر، الگوریتم تخمین و جبران‌سازی حرکت که از بخشهای زمان‌بر و با هزینه محاسباتی بالا در فشرده‌سازی ویدئو است، سریعتر انجام خواهد شد.

دلیل بروز این مشکل اینست که الگوریتم تخمین حرکت مجبور است در محدوده جستجو، بلاکی را که کمترین اختلاف متوسط شدت روشنایی MAD را با بلاک مورد نظر دارد انتخاب نماید. حال اگر در این محدوده، بلاکی با متوسط سطح روشنایی برابر یا نزدیک به بلاک مورد نظر وجود نداشته باشد، تخمین مناسبی انجام نخواهد شد و در فریم جبران سازی شده، این اختلاف شدت روشنایی قابل رویت خواهد بود. در الگوریتم پایه چون تفاضل ارسال می شود، مشکلی بوجود نخواهد آمد، چون این میزان اختلاف توسط فریم تفاضل، جبران می شود. اما در روش پیشنهادی که فریم تفاضلی فرستاده نمی شود، این مساله وجود خواهد داشت.

برای رفع این مشکل، علاوه بر جبران سازی حرکت، باید جبران سازی را برای شدت روشنایی نیز انجام داد و برای منطقی تر بودن عملیات تخمین، بهتر است اصلاح شدت روشنایی قبل از تخمین حرکت انجام شود. برای این منظور به صورت زیر عمل می شود. برای یافتن شبیه ترین بلاک به بلاک مورد نظر، ابتدا تفاضل بین بلاک موجود در فریم مرجع، b_1 و بلاک متناظر در فریم جاری، b_{2j} ، محاسبه شده و میانگین آن بعنوان پارامتر MD در نظر گرفته می شود.

$$MD = \text{mean}(b_{2j} - b_1) \quad (2)$$



شکل ۳: دیاگرام ساختار متوالی در روش پیشنهادی

از آنجا که مقدار MD تفاوت شدت روشنایی متوسط بین دو بلاک را بیان می کند، برای اصلاح شدت روشنایی بلاک b_{2j} ، جزء صحیح MD به تمامی پیکسلهای بلاک b_1 اضافه می شود.

حال معیار میانگین قدرمطلق اختلاف شدت روشنایی یا MAD بین بلاک جاری و بلاک اصلاح شده محاسبه می شود. این مراحل برای تمامی بلاکهای b_{2j} در محدوده جستجو انجام شده و بلاکی که کمترین MAD را با بلاک b_1 داشته باشد، به عنوان بهترین تخمین در نظر گرفته شده و علاوه بر میزان جابجایی، مقدار اختلاف شدت روشنایی متوسط یا همان جزء صحیح MD نیز برای کدگشا ارسال خواهد شد. با انجام این تغییر، عملیات تخمین حرکت با دقت بیشتر و بصورت منطقی تری انجام خواهد شد. همچنین چون میزان اختلاف میانگین شدت روشناییها نیز ارسال می شود، مشکل موجود در قسمت قبل نیز برطرف خواهد شد. شکل ۴ تاثیر این تغییر را در بهبود کیفیت

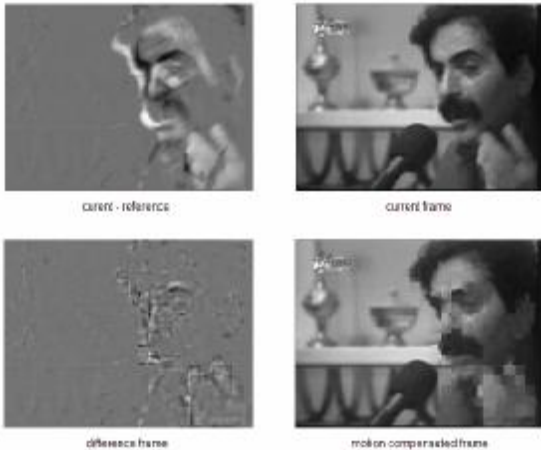
فریم جبران سازی شده مرحله $i-1$ بعنوان فریم مرجع قبلی در نظر گرفته شده و با فریم جاری مقایسه می شود. در این مرحله تخمین و جبران سازی حرکت با بلاکهای کوچکتری انجام خواهد شد. با ادامه این دیاگرام تا سطح پیکسلی، جبران سازی حرکت طوری انجام می شود که فریم جبران سازی شده، بسیار شبیه به فریم جاری بدست خواهد آمد. بنابراین دیگر نیازی به ارسال فریم تفاضل نخواهد بود.

میزان شباهت فریم جبران سازی شده نهایی با فریم جاری، با تعیین حدهای آستانه Th_i مشخص می شود. این آستانه ها میزان حداکثر اختلاف مجاز بین فریم مرجع و جاری را تعیین می کنند. با کوچک انتخاب کردن مقدار آنها، اختلافهای کوچک بین دو فریم نیز در نظر گرفته شده و باید جبران شوند، اما انتخاب مقادیر بزرگ باعث صرف نظر کردن از حرکتها و اختلافهای کوچک خواهد شد و الگوریتم را سریعتر پیش خواهد برد. برای اینکه بتوان شباهت بیشتری را برای فریم جبران سازی شده با حجم محاسباتی کمتر ایجاد نمود، بهتر است بجای استفاده از آستانه های ثابت، مقدار آنها را در هر مرحله بطور مناسب انتخاب نمود. به این ترتیب که در مراحل که بازای یک حد آستانه مشخص، عملیات محاسباتی کمتری انجام می شود، حد آستانه، مقدار کوچکتری انتخاب شود تا تخمین حرکت با سخت گیری بیشتری انجام شود و برای مراحل با حجم عملیاتی بالا، مقدار این آستانه بزرگتر انتخاب گردد.

از آنجایی که در مراحل اولیه، اندازه بلاکها بزرگتر است و تعداد عملیات تخمین حرکت کمتر می باشد و همچنین اینکه صحنه های با حرکت کم معمولا سطح بزرگتری را اشغال می کنند، بهتر است مقدار حد آستانه، کوچکتر انتخاب شود. با پیشروی در دیاگرام، مقدار حد آستانه بزرگتر انتخاب می شود.

۳-۳- جبران سازی شدت روشنایی

با استفاده از دو تغییری که توضیح داده شد، می توان مقدار پارامترها را طوری تنظیم نمود که کیفیت مطلوب را با تعداد بیت و حجم محاسبات مناسب ارائه دهد، بدون اینکه نیازی به ارسال فریم تفاضل باشد. اما با اعمال این الگوریتم به یک سری متوالی از فریمها مشاهده می شود که در برخی نواحی از فریمهای جبران سازی شده، میزان شدت روشنایی با فریم اصلی تفاوت زیادی پیدا کرده است.



شکل ۵: جبران سازی حرکت با حد آستانه ۵ و اندازه بلاک ۱۶×۱۶



شکل ۶: جبران سازی حرکت با حد آستانه ۵ و اندازه بلاک ۱×۱



شکل ۷: جبران سازی با ساختار هرمی و با حد آستانه ثابت ۵

تصویر و رفع مشکل تغییر شدت روشنایی برخی نواحی نشان می‌دهد. با ارسال میزان اختلاف میانگین شدت روشنایی برای کدگشا به همراه بردارهای حرکت، با اینکه تعداد بیشتری داده برای ارسال ایجاد می‌شود اما از آنجا که تخمین حرکت با اصلاح شدت روشنایی بلاکها انجام می‌شود، تخمین مناسبتری انجام شده و تعداد کمتری از بلاکها در مرحله بعدی نیاز به جبران سازی خواهند داشت.



شکل ۴: تاثیر جبران سازی شدت روشنایی در بهبود کیفیت تصویر

۴- نتایج آزمایشات

شکلهای ۵ و ۶ و جدول ۱ نتیجه جبران سازی با بلاکهای ۱۶×۱۶ و ۱×۱ را نشان می‌دهند. مشاهده می‌شود که در فریم جبران سازی شده با اندازه ۱۶×۱۶، کیفیت بسیار پایینی بدست می‌آید و بلاکهای مربعی و مجزا در تصویر قابل رویت هستند، در صورتیکه با جبران سازی حرکت در سطح پیکسلی اثری از بلاکهای مربعی مجزا مشاهده نمی‌شود.

معیار حداقل تعداد بیت مورد نیاز برای فشرده سازی فریم تفاضل، تنها برای مقایسه کارایی عملیات جبران سازی در جداول آورده شده است و در روش پیشنهادی، این داده‌ها برای کدگشا ارسال نمی‌شوند. هر چه مقدار آن کمتر باشد به این معنی است که انتروپی فریم تفاضل کاهش یافته و جبران سازی بهتری انجام شده است. حداقل تعداد بیت مورد نیاز برای فشرده سازی فریم تفاضل بین فریم جاری و مرجع، ۷۱۰۶۸ بیت بدست آمد.

برای بررسی تاثیر مقدار حد آستانه در ساختار متوالی پیشنهادی، نتایج آزمایش با دو حد آستانه ۵ و ۱۵ در شکلهای ۷ و ۸ نشان داده شده است. بازای حد آستانه کوچکتر، کیفیت فریم جبران سازی شده نهایی بهبود یافته است، اما در عوض حجم محاسباتی بالایی مورد نیاز است. شکل ۹ نتایج بدست آمده بازای حدهای آستانه مختلف در ساختار متوالی هرمی، از ۲ برای مرحله اول تا ۱۶ در مرحله نهایی را نشان می‌دهد. همچنین حجم محاسبات و تعداد بایتهای ایجاد شده برای فشرده سازی بردارهای حرکت، بازای این آستانه‌های مختلف در جدول ۲ آورده شده‌اند. در شکلهای ۷ تا ۹، d و b_s بترتیب، اندازه بلاک و طول ناحیه جستجو می‌باشند.

می‌شود که ساختار متوالی توانسته است با تعداد عملیات تخمین کمتر و تعداد بیت کمتر، کیفیت جبران‌سازی بهتری را ارائه داده و مقدار *MAD* را کاهش دهد.

برای درک بهتر تاثیر جبران‌سازی شدت روشنایی در بهبود کارایی عملیات جبران‌سازی، روند ذکر شده در شکل ۹، با اعمال این تغییر، انجام و نتیجه آن در شکل ۱۰ نشان داده شده است. همچنین نتایج بدست آمده از این دو آزمایش در جدول ۳ مقایسه شده است.

نتایج جدول ۳ نشان می‌دهد که استفاده از جبران‌سازی شدت روشنایی با اضافه شدن تعداد کمی به حداقل بایت مورد نیاز، تخمین مناسبتری را ارائه می‌نماید. زیرا نه تنها تعداد عملیات تخمین کمتری صورت می‌گیرد و سرعت جبران‌سازی افزایش می‌یابد، بلکه مقدار *MAD* نیز کاهش خواهد یافت و این به معنی تخمین بهتر و جبران‌سازی مناسبتر می‌باشد.



شکل ۱۰: جبران‌سازی حرکت بر طبق روال استفاده شده در شکل ۸ و با اعمال جبران‌سازی شدت روشنایی

جدول ۳: بررسی تاثیر استفاده از جبران‌سازی شدت روشنایی

حد آستانه در هر مرحله	جبران‌سازی شدت روشنایی	تعداد عملیات تخمین حرکت	حد اقل تعداد بایت		<i>MAD</i>
			برای فشرده‌سازی بردارهای حرکت و روشنایی	برای فشرده‌سازی فریم تفاضل	
۱۶-۱۳-۹	خیر	۴۶۱۰	۶۳۹۶	۴۸۶۹۲	۲/۰۸۹
۶-۲	بلی	۳۰۷۴	۷۴۵۵	۴۶۵۷۹	۱/۸۶۲

۵- جمع بندی

از سه تغییر پیشنهاد شده در این مقاله، دو تغییر اول و سوم می‌توانند به تنهایی نیز به الگوریتم پایه اعمال شده و نتایج را بهبود بخشند. تغییر اول باعث سریعتر شدن عملیات تخمین حرکت شده و تغییر سوم با



شکل ۸: جبران‌سازی با ساختار هرمی و با حد آستانه ثابت ۱۵



شکل ۹: جبران‌سازی با ساختار هرمی و با حد آستانه‌های

۲، ۶، ۹، ۱۳ و ۱۶ بترتیب برای مراحل ۱ تا ۵

جدول ۱: مقایسه جبران‌سازی با اندازه بلاکهای مختلف

اندازه بلاک	تعداد عملیات تخمین حرکت	حد اقل تعداد بایت		<i>MAD</i>
		برای فشرده‌سازی بردارهای حرکت	برای فشرده‌سازی فریم تفاضل	
۱	۳۹۶۷۹	۵۳۷۲۸	۳۶۵۴۶	۱/۲۱۱
۱۶	۲۴۰	۲۱۵	۶۰۱۰۱	۴/۵۶۳

جدول ۲: مقایسه نتایج بدست آمده برای حد آستانه‌های مختلف

حد آستانه در هر مرحله	تعداد عملیات تخمین حرکت	حد اقل تعداد بایت		<i>MAD</i>
		برای فشرده‌سازی بردارهای حرکت	برای فشرده‌سازی فریم تفاضل	
۳-۳-۳-۳	۲۱۱۷۷	۳۹۴۱۳	۲۹۳۱۲	۰/۷۲۸
۵-۵-۵-۵	۱۹۶۵۷	۲۷۸۴۷	۳۸۳۶۷	۱/۲۳۴
۱۵-۱۵-۱۵-۱۵	۴۲۳۳	۷۲۰۵	۵۰۸۱۹	۲/۳۴۷
۲-۶-۹-۱۳-۱۶	۴۶۱۰	۶۳۹۶	۴۸۶۹۲	۲/۰۸۹

شکل ۹ و نتایج جدول ۲ نشان می‌دهند که با انتخاب حد آستانه‌های مناسب برای هر مرحله، تعداد عملیات تخمین کمتری انجام خواهد شد، اما کیفیت فریم جبران‌سازی شده در حد مطلوب بدست خواهد آمد. همچنین با مقایسه سطر اول در جدولهای ۱ و ۲ مشاهده

افزایش دقت در تخمین، باعث خواهد شد که انتروپی فریم تفاضلی کاهش یافته و تعداد بیت کمتری برای ارسال آن در روش پایه مورد نیاز باشد. همچنین نتایج جدول ۲ به خوبی نشان می‌دهند که استفاده از ساختار متوالی و کاهش اندازه بلاکها در مراحل انتهایی، می‌تواند کارایی جبران‌سازی را بهبود بخشد و ضمن افزایش سرعت، تعداد بیت کمتری نیز مورد نیاز خواهد بود.

با بررسی دو معیار MAD و حداقل بایت مورد نیاز برای فشرده‌سازی فریم تفاضل در جدول ۳، تاثیر جبران‌سازی شدت روشنایی در کارایی عملیات جبران‌سازی حرکت، به روشنی قابل مشاهده است، زیرا با اعمال جبران‌سازی شدت روشنایی، مقدار هر دو پارامتر کاهش یافته است.

مراجع

- [1] [1] Koga T., Iinuma K., Hirano A., Iijima Y., and Ishiguro T., "Motion compensated interframe coding for video conferencing," in Proc. Nat. Telecommun. Conf., New Orleans, LA, pp. G5.3.1-G5.3.5, Nov. 1981.
- [2] [2] Lee J. H., Ra J. B., "Efficient motion estimation using edge-based binary block-matching and refinement based on motion vector correlation", ICIP (2), pp. 957-960, 2001.
- [3] [3] Jain J. R., Jain A. K., "Displacement measurement and its application in interframe image coding," IEEE Trans. Commun., Vol. COM-29, No. 12, pp. 1799-1808, Dec. 1981.
- [4] [4] Liu L. K., Feig E., "A block-based gradient descent search algorithm for block-based motion estimation in video coding," IEEE Trans. Circuits Syst. Video Technol., Vol. 6, No. 4, pp. 419-422, Aug. 1996.
- [5] [5] Chen L. G., Chen W. T., Jehng Y. S., Chuieh T. D., "An efficient parallel motion estimation algorithm for digital image processing," IEEE Trans. Circuits Syst. Video Technol., Vol. 1, No. 4, pp. 378-384, Dec. 1991
- [6] [6] Chan Y. L., Siu W. C., "Adaptive multiple-candidate hierarchical search for block matching algorithm," IEE Electron. Lett., Vol. 31, No. 19, pp. 1637-1639, Sep. 1995.
- [7] [7] Cheung C. K., Po L. M., "A hierarchical block motion estimation algorithm using partial distortion measure," in Proc. ICIP'97, Vol. 3, pp. 606-609, 1997.
- [8] [8] Zheng W., Ahmad I., Liou M. L., "Adaptive motion search with elastic diamonds for MPEG-4 video encoding," in Proc. Int. Conf. Image Processing, Thessaloniki, Greece, pp. 377-380, Oct. 2001.
- [9] [9] Zhu S., Ma K. K., "A new diamond search algorithm for fast block matching," IEEE Trans. Circuits Syst. Video Technol., Vol. 9, No. 2, pp. 287-290, Feb. 2000.