

الگوریتم جدید انتخاب مُد داخل فریمی سریع در استاندارد کدگذاری ویدیویی H.264/AVC

مجید محمدی

**
مهدی جعفری

*
مهناز نژادعلی

* کارشناس ارشد، مهندسی برق و الکترونیک، دانشگاه آزاد اسلامی، واحد علوم و تحقیقات
** استادیار، گروه برق، دانشگاه آزاد اسلامی، دانشکده فنی، کرمان
*** دانشیار، گروه برق و کامپیوتر، دانشگاه شهید بهشتی، دانشکده مهندسی برق و کامپیوتر، تهران
تاریخ دریافت: ۹۲/۰۳/۰۴ تاریخ پذیرش: ۹۴/۱۱/۲۰

چکیده

با افزایش روزافزون ارتباطات ویدیویی در دستگاه‌های قابل حمل و کاربردی، طراحی کدکننده‌ها با میزان پیچیدگی کم و عملکرد بالا مورد نیاز است. استاندارد H.264/AVC کارایی فشرده‌سازی بالاتری را نسبت به استانداردهای قبلی ارائه می‌دهد. اما این استاندارد با بکارگیری چندین تکنیک کدگذاری قدرتمند، پیچیدگی کدگذار را بطور قابل ملاحظه‌ای افزایش می‌دهد. این مقاله الگوریتم جدیدی برای کاهش پیچیدگی کدگذار H.264/AVC ارائه می‌دهد. روش پیشنهادی از ماسک‌های جهت‌دار ساده، مُد بلوک‌های همسایه و تشخیص مُد تخمین داخلی ۴×۴ و/یا ۱۶×۱۶ با تعیین پارامترهای کوانتیزه به منظور انتخاب سریع مُد در پیش‌بینی به روش داخل فریمی استفاده می‌کند. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی زمان کدگذاری را ماکزیمم ۲۹٪ درصد کاهش می‌دهد در حالی که تاثیر ناچیزی در کیفیت تصویر و میزان PSNR دارد.

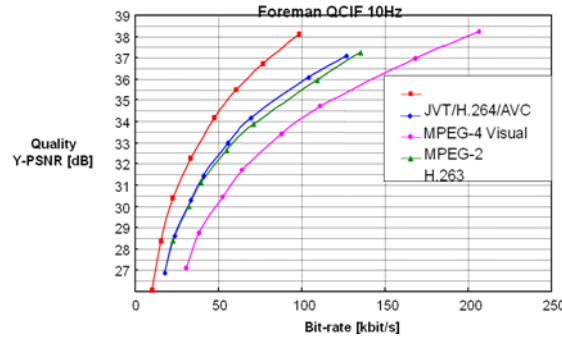
واژه‌های کلیدی: کدگذاری ویدیو، H.264/AVC، پیش‌بینی، داخل فریمی، RDO

۱- مقدمه

استاندارد H.264/AVC می‌تواند کیفیت تصویر خوبی را در نرخ داده‌ای که اساساً پایین‌تر از آنچه استانداردهای قبلی ارائه دادند بدون پیچیده و گران کردن سیستم فراهم آورد. این استاندارد دارای کارایی بسیار بالاتری نسبت به استانداردهای پیشین می‌باشد و به دلایل افزایش بازده فشرده‌سازی (نرخ بیت متوسط آن تا ۵۰٪ نسبت به H.263، MPEG-4 و MPEG-2 صرفه‌جویی می‌کند)، سازگاری با محدودیت تاخیر زمانی، مقاومت در مقابل خطا و آسانی کارکرد

از آنجائیکه کاربردهای جدید چندرسانه‌ای (استفاده از انواع مختلف شبکه‌ها) به سرعت در حال رشد می‌باشد، کدگذاری ویدیو به کارایی بیشتر و ویژگی‌های تازه نیازمند است. استاندارد کدگذاری ویدیو H.264/AVC در سال ۲۰۰۳ و به طور مشترک توسط تیم‌های ویدیویی ISO/IEC MPEG و ITU_T VCEG تحت عنوان استاندارد بین-المللی کدگذاری ویدئو پیشرفته با نام 14496-10 (MPEG-4 part 10 یا H.264/AVC) ارائه شده است

[1]



شکل ۱- مقایسه کدک‌های H.263, MPEG-4, MPEG-2 با H.264 [2]

ها از اطلاعات داخل یک فریم بیشتر است روش داخل فریمی ترجیح داده می‌شود و تعداد بیت کمتری ایجاد می‌کند. همچنین ماکروبلوک‌هایی که به صورت داخل فریمی کدگذاری می‌شوند در مقابل خطای کانال مقاوم‌تر هستند و مصونیت را در مقابل خطای کانال افزایش می‌دهند [2].

پیشگویی داخل فریمی از مهمترین عوامل برتری تکنیک کدینگ H.264/AVC نسبت به دیگر روش‌های فشرده سازی استاندارد است که از تشابهات مکانی جهت‌دار برای کاهش افزونگی استفاده می‌کند. تکنیک بهینه‌سازی نرخ اعوجاج (RDO) در این استاندارد برای انتخاب بهترین مُد پیش‌بینی داخل فریمی و دستیابی به بهترین مُد کدکردن و بالاترین بازدهی بکار گرفته می‌شود [1]. کدگذار در کدکردن بلوک موردنظر، برای یافتن بهترین حالت نرخ اعوجاج (RD)، باید همه مُدهای ممکن برای کدینگ داخلی را تست کند و ارزش‌های RD آنها را محاسبه نماید تا مُدی که کمترین اعوجاج را ایجاد کند، برگزیند. اما تعیین مقادیر ارزش RD برای نمونه‌های روشنایی (LUMA) و رنگ (CHROMA) در یک ماکروبلوک، نیازمند محاسبات بسیار بوده و منجر به افزایش پیچیدگی می‌گردد [3] که کارکرد این استاندارد را برای کاربردهای بلادرنگ با مشکل مواجه می‌کند.

تلاش‌های زیادی برای توسعه الگوریتم‌های پیش‌بینی داخل فریمی سریع و کاهش زمان کدگذاری در این استاندارد انجام شده است. پان و همکاران [4، 5، 6] الگوریتم‌های برآورد مُد داخلی با استفاده از نمودار ستونی (هیستوگرام) آشکارسازی لبه و شناسایی لبه محلی با روش پیش‌پردازش را پیشنهاد کردند، که در آن اندازه‌گیری جهت لبه بلوک به منظور کاهش تعداد مُدهای احتمالی و کاهش پیچیدگی

در انواع مختلف شبکه‌ها، توجه زیادی را به خود جلب کرده است. شکل (۱) کیفیت این کدکننده را در مقایسه با سایر کدک‌ها و در نرخ بیت‌های مختلف نشان می‌دهد. در این استاندارد مزایای روش‌های مختلف استانداردهای کدگذاری قبلی بکار گرفته شده و در یک پروسه با هم ترکیب شده‌اند؛ پیش‌بینی داخل و بین‌فریمی، تبدیل با عنصرهای اعداد صحیح، کدگذاری طول-صفر و کدگذاری آنتروپی منجر به کاهش افزونگی مکانی، زمانی و آماری در این استاندارد گردیده است [2].

از عواملی که کدک H.264 را از استانداردهای قبلی متمایز ساخته جبران حرکت در ماکروبلوک‌های 16×16 پیکسلی با ابعاد مختلف است و این استاندارد برای کدگذاری بیشترین استفاده را از مُد تخمین/جبران حرکت یا پیش‌بینی بین فریمی می‌برد. دقت جبران حرکت در این استاندارد در حالت عادی یک چهارم پیکسل است و می‌تواند تا یک هشتم پیکسل با پیچیدگی بیشتر افزایش یابد. کدگذار H.264 از بلوک‌های با سایز کوچک‌تر برای بخش‌های متحرک تصویر و از بلوک‌های با سایز بزرگ‌تر برای پس زمینه‌ها استفاده می‌کند تا کارایی کدینگ و کیفیت را همزمان بالا ببرد.

اما گاهی بهتر است یک ماکروبلوک به جای مُد بین فریمی به صورت داخل فریمی کدگذاری شود. برای این کار واریانس خود بلوک را با واریانس تفاوت آن با فریم قبلی مقایسه کرده و کمترین انرژی را مبنا قرار می‌دهیم. برای انرژی‌های زیاد (واریانس بالا) تفاوتی در انتخاب مُد داخل/بین فریمی وجود نداشته ولی در انرژی‌های کم نوع بین فریمی ترجیح داده می‌شود. لذا در تصاویر ویدیویی که در اثر تغییر صحنه و یا حرکت‌های سریع تفاوت بین فریم-

سریع مبتنی بر جهت بافت و زمینه تصویر در [14] ارائه شده است که تنها زیر مجموعه ای از مُدهای داخلی را مطابق با جهت بافت تصویر شناسایی شده، تست می‌نماید و می‌تواند تا ۶۰٪ زمان کدگذاری را کاهش دهد.

در این مقاله الگوریتم ساده‌ای برای انتخاب مُد پیش‌بینی داخل فریمی ۴×۴ در نظر گرفته شده است. این الگوریتم با استفاده از ماسک‌های جهت‌دار و مُد بلوک‌های همسایه، بهترین مُد کاندید برای تخمین بلوک فعلی را بدست می‌آورد. با این روش می‌توان تعداد مُدهای RD در بلوک‌های ۴×۴ را از ۹ مُد، در بدترین شرایط به ۴ مُد و در بهترین شرایط به یک مُد کاندید کاهش داد. نتایج شبیه‌سازی نشان می‌دهد که این روش می‌تواند به میزان قابل توجهی زمان کدگذاری را کاهش دهد.

ادامه این مقاله بصورت زیر تنظیم گردیده است: در بخش ۲ پیش‌بینی داخل فریمی استاندارد H.264/AVC را بازبینی می‌کنیم. در بخش ۳ الگوریتم جدید مُد داخل فریمی سریع پیشنهادی مبتنی بر ماسک‌های جهت‌دار، مُد بلوک‌های همسایه و تشخیص مُد تخمین داخلی ۴×۴ و/یا ۱۶×۱۶ با تعیین پارامترهای کوانتیزه معرفی می‌گردد. نتایج شبیه‌سازی روش پیشنهادی در بخش ۴ ارائه می‌شود و در نهایت نتیجه‌گیری و پیشنهادات در بخش ۵ آورده شده است.

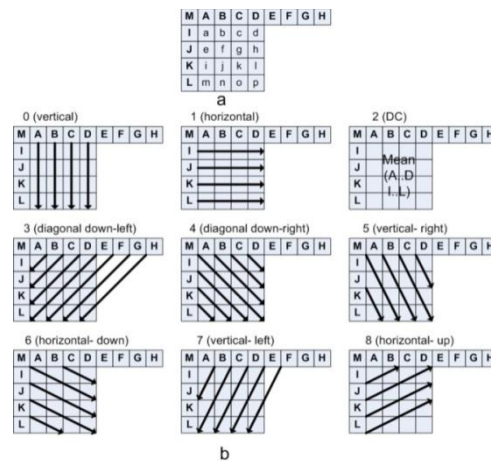
۲- پیش‌بینی داخل فریمی در استاندارد H.264/AVC

از آنجائیکه در یک فریم تصویر میزان افزونگی بالاست، مُد داخل فریمی در کاهش بیت‌های افزونگی موردنیاز است در واقع پیش‌بینی داخل فریمی برای کاهش افزونگی مکانی در تصویر بکار برده می‌شود. در این روش ماکروبلوک فعلی از پیکسل‌های مجاور، ماکروبلوک‌های بالا و/یا چپ در انتخاب مُد تخمین و افزایش بازده کدگذاری استفاده می‌کند. تفاوت بین ماکروبلوک اصلی و ماکروبلوک کدگشایی شده توسط بهترین مُد تخمین کدگذاری می‌شود. لذا تعداد بیت‌های موردنیاز در کدگذاری نسبت به حالتی که تبدیل روی ماکروبلوک اصلی انجام گردد به مراتب پایین‌تر خواهد بود. مجموعه حالت‌های کدگذاری برای اطلاعات روشنایی در روش داخل فریمی، ۴×۴ و ۱۶×۱۶ و در تخمین رنگ، ۸×۸

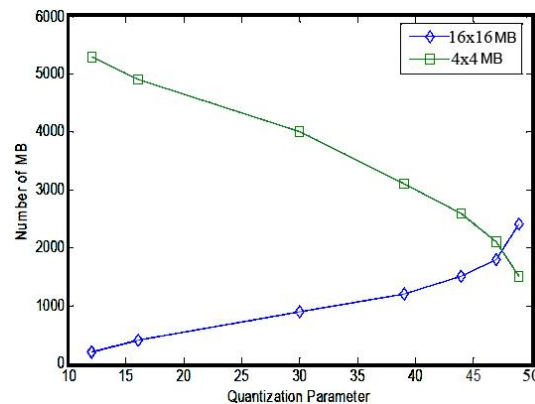
انجام می‌گیرد. هرچند این روش برای آشکارسازی راستای لبه و دسته‌بندی در یک راستای محدود، نیازمند عملیات اضافی بوده که باعث اتلاف زمان کدگذاری می‌گردد. یک الگوریتم تصمیم‌گیری مُد داخلی سریع برای پیش‌بینی داخلی H.264 در مرجع [7] ارائه شده است. این الگوریتم با استفاده از ویژگی‌های ماکروبلوک و تخمین یکنواختی و همگن بودن آن یکی از مُدهای ۴×۴ و ۱۶×۱۶ را برای تعیین بهترین مُد پیش‌بینی برمی‌گزیند. این الگوریتم ۳۰ تا ۶۰ درصد محاسبات را کاهش داده در حالیکه عملکرد PSNR و نرخ بیت با استاندارد H.264 یکسان می‌باشد. الگوریتم سریعی برای انتخاب مد پیش‌بینی داخلی بهینه با بکار بردن ماسک‌های ساده جهت‌دار و اطلاعات مد بلوک همسایه، در مرجع [8] وجود دارد که تا ۷۰٪ زمان کدگذاری را کاهش می‌دهد. روشی مبتنی بر خصوصیات آماری برای پیش‌بینی مُد داخلی روشنایی در مرجع [9] ارائه شده است که ۴۵٪ در زمان کدگذاری صرفه جویی می‌کند. مرجع [10] با پیشنهاد دو تکنیک "تصمیم مُد پرش زود هنگام" و "تصمیم مُد داخلی انتخابی" پیچیدگی کدگذاری را کاهش داده و سرعت کدگذاری را به طور متوسط ۳۰٪ بهبود می‌بخشد. در مرجع [11]، یک الگوریتم سریع بر پایه انتخاب نوع بلوک و اطلاعات لبه محلی که از محاسبه پارامترهای ویژگی لبه و تکنیک‌های نمونه‌برداری از عملیات انطباقی بدست می‌آید، معرفی شده است. این روش با افت بسیار کمی در مقدار PSNR، زمان کدگذاری را بیش از ۷۱٪ کاهش می‌دهد. روشی مبتنی بر بردار مرکز ثقل بلوک در مرجع [12] پیشنهاد شده است که جهت این بردار برای انتخاب جهت پیش‌بینی داخلی استفاده می‌شود. این الگوریتم با اتخاذ روشی هوشمند تعداد مُدهای کمتری که در محاسبات RD موردنیاز است بر می‌گزیند و فقط با محاسبه مقدار زاویه مرکز ثقل بهترین مُد پیش‌بینی داخل فریمی را انتخاب می‌کند. مرجع [13] تکنیک انتخاب مُد پیش‌بینی داخل فریمی را ارائه می‌کند که در آن محتوای بلوک و پیکسل‌های کناری بازسازی شده به منظور تسهیل روند کدگذاری بکار برده می‌شود. این روش می‌تواند به طور موثر مُدهای با احتمال پایین در محاسبات نرخ اعوجاج (RD) را حذف نماید. طرح تصمیم‌گیری مُد داخلی

زیرا تعداد مُدهای تخمین تأثیری در بازده کدگذاری ندارد و فقط اطلاعات سربار برای مشخص کردن جهت تخمین را بالا می‌برد. از این رو در این قسمت‌های تصویر در استاندارد H.264، کل ماکروبلوک (۱۶×۱۶) از ۴ نوع تخمین که عبارت‌اند از DC، افقی، عمودی و صاف استفاده می‌کند. در این استاندارد برای تخمین بلوک‌های رنگ، از تکنیک مشابه ۱۶×۱۶ و با ۴ تخمین برای دو بلوک رنگ پیشنهاد می‌گردد.

می‌باشد. استاندارد اصلی H.264 هشت جهت تخمین به اضافه یک مُد DC (مُد ۲) را برای کدگذاری داخل فریمی بلوک‌های روشنایی ۴×۴ استفاده می‌کند. جهت‌های تخمین همراه با بلوک ۴×۴ و پیکسل‌های مجاور آن در شکل (۲) نشان داده شده‌اند. جهت پیکان‌ها در این شکل جهت تخمین برای کدگذاری بلوک ۴×۴ را نشان می‌دهند. در قسمت‌هایی از تصویر که جزئیات زیادی وجود ندارد (نواحی صاف تصویر)، لزومی ندارد که از ۹ مُد تخمین استفاده شود



شکل ۲- (a) بلوک ۴×۴ و پیکسل‌های مجاور (b) نه مُد تخمین در بلوک ۴×۴ روشنایی [15]



شکل ۳- تعداد ماکروبلوک‌های کدشده داخلی ۴×۴ و ۱۶×۱۶ در پارامترهای کوانتیزه مختلف [16]

۳-۱-۳- تصمیم مُد داخلی در بلوک‌های ۴×۴

استاندارد H.264/AVC در تخمین داخلی بلوک‌های ۴×۴ روشنایی، ۹ مُد تخمین بصورت نشان داده در شکل (۲) را بکار می‌برد. در نرم‌افزار مرجع، پیکسل‌ها در جهت تخمین یکسان مقادیر یکسانی دارند؛ برای مثال در مُد ۰ (مُد عمودی) همه پیکسل‌های تخمین زده شده برای موقعیت-های a, e, i و m باید دارای مقادیر یکسان و مساوی با مقدار پیکسل A طبق شکل (۲) باشند. پس از محاسبات RDO اگر مُد ۰ به عنوان بهترین مُد پیش‌بینی برگزیده شود، در آن صورت مقادیر پیکسل‌های تخمین زده شده در امتداد جهت عمودی یکسان بوده و این در اصل نشان‌دهنده آنست که در بلوک اصلی، مقادیر پیکسل‌ها در این موقعیت-ها بسیار شبیه به هم هستند. بنابراین در این مقاله روش تخمین بهترین مُد کاندید بر اساس محاسبات NSAD یا مجموع نرمال شده قدرمطلق تفاوت‌ها (Normalized Sum of Absolute Differences) با انتخاب موقعیت برخی پیکسل‌ها در بلوک اصلی طبق کد نوشته شده زیر می‌باشد.

$$\text{NSAD } 0 = (\text{abs}(a-m) + \text{abs}(b-n) + \text{abs}(c-o) + \text{abs}(d-p)) \gg 2;$$

$$\text{NSAD } 1 = (\text{abs}(a-d) + \text{abs}(e-h) + \text{abs}(i-l) + \text{abs}(m-p)) \gg 2;$$

$$\text{NSAD } 3 = (\text{abs}(b-e) + \text{abs}(d-m) + \text{abs}(l-o)) / 3;$$

$$\text{NSAD } 4 = (\text{abs}(a-p) + \text{abs}(i-n) + \text{abs}(c-h)) / 3;$$

$$\text{NSAD } 5 = (\text{abs}(a-j) + \text{abs}(b-k) + \text{abs}(c-l)) / 3;$$

$$\text{NSAD } 6 = (\text{abs}(a-g) + \text{abs}(e-k) + \text{abs}(i-o)) / 3;$$

$$\text{NSAD } 7 = (\text{abs}(b-i) + \text{abs}(c-j) + \text{abs}(d-k)) / 3;$$

$$\text{NSAD } 8 = (\text{abs}(e-c) + \text{abs}(i-g) + \text{abs}(m-k)) / 3;$$

چون مُد 2 (DC) از میانگین مقادیر پیکسل‌های A تا M تخمین زده می‌شود و بدون جهت می‌باشد رابطه (۳) برای تخمین مُد DC در نظر گرفته شده و با DDC نشان داده می‌شود.

$$DDC = \sum_{i=0}^{15} |m-p_i| \quad (3)$$

برای انتخاب بهترین مُد تخمین از میان مُدهای فوق این استاندارد از تکنیک RDO استفاده می‌کند و مُدی با کمترین ارزش RD به عنوان بهترین مُد برگزیده می‌شود. روش RDO مبتنی بر تابع لاگرانژ بوده و دادوستدی بین نرخ بیت و اعوجاج در آن مطرح می‌باشد. ارزش RD برای تعیین بهترین مُد تخمین داخل فریمی از معادله (۱) بدست می‌آید:

$$J = D + \lambda R \quad (1)$$

که در آن D اعوجاج، R نرخ بیت و λ ضریب لاگرانژین می‌باشد و از رابطه (۲) محاسبه می‌گردد:

$$\lambda = 0.85 \times 2^{(QP-12)/3} \quad (2)$$

λ کوچکتر به معنای اعوجاج کمتر و نرخ بیت بیشتر و λ بزرگتر به معنای اعوجاج بیشتر و نرخ بیت کمتر است. R تعداد بیت مربوط به هر مُد و D از مجموع اختلاف مربعات بین بلوک اصلی و بلوک بازسازی شده محاسبه می‌گردد. بنابراین تعداد ترکیبات ممکن انتخاب مُد برای اجزای روشنایی و رنگ در یک ماکروبلوک 8×8 خواهد بود که $N8 \times (N4 \times 16 + N16) = 4 \times (9 \times 16 + 4) = 592$ رنگ، 4×4 و 16×16 روشنایی می‌باشند. بنابراین برای انتخاب بهترین مُد تخمین در مُد داخل فریمی در یک ماکروبلوک نیاز به ۵۹۲ محاسبه RDO در کدگذار H.264 خواهد بود که به شدت پیچیدگی کدگذار را افزایش می‌دهد.

۳- الگوریتم تصمیم مُد داخلی سریع

در تخمین نمونه‌های روشنایی در مُد داخلی و پارامترهای کوانتیزه معمول (۲۰۴۵)، احتمال سایز بلوک ۴×۴ بیشتر از سایز بلوک ۱۶×۱۶ می‌باشد؛ همانطور که در شکل (۳) نشان داده شده است. بنابراین در این مقاله با تشخیص سریع مُد تخمین داخلی ۴×۴ در QP‌های پایین و ۱۶×۱۶ در QP‌های بالا بطور قابل ملاحظه‌ای سرعت کدگذاری افزایش می‌یابد [16].

مرحله ۲-۲: اگر DDC کوچکتر از آستانه T1 باشد، آنگاه روش RDO در نهایت شامل ۴ مُد کاندید است. یک مُد با کمترین مقدار محاسبه شده NSAD، دو مُد از بلوک‌های مجاور و مُد DC [18].

۳-۲- تصمیم مُد داخلی در بلوک‌های ۱۶×۱۶ روشنایی و بلوک‌های ۸×۸ رنگ

در این مقاله محاسبات RDO و انتخاب مُد داخلی در بلوک‌های ۱۶×۱۶ روشنایی و بلوک‌های ۸×۸ رنگ، همان روش بکار برده شده در نرم افزار مرجع JM 18.4 می‌باشد.

۴- نتایج شبیه‌سازی روش پیشنهادی

الگوریتم پیشنهادی در JM18.4 که توسط JVT ارائه گردید پیاده‌سازی شد [19]. آزمایش روی فریم‌های ویدیویی Coastguard, Silent, Forman, Mobile, Bus و Tempete با ابعاد CIF و QCIF انجام گردید و شبیه‌سازی با پارامترهای کوانتیزه ۱۶، ۲۸، ۳۲، ۴۴ و بر روی فریم‌های نوع I (تعداد ۱۰۰ فریم در تصاویر ویدیویی QCIF و ۳۰ فریم در تصاویر CIF) انجام شد. سخت‌افزار استفاده شده در شبیه‌سازی روش پیشنهادی، کامپیوتر Intel Core i3 CPU با حافظه 2GB و سرعت 2.13 GHz و ویندوز 7 می‌باشد. برای سنجش کارایی، تفاوت PSNR (PSNR)، نرخ-بیت (ABR) و درصد زمان صرفه‌جویی روش پیشنهادی در مقایسه با نرم‌افزار مرجع JM 18.4 توسط روابط (۵)، (۶) و (۷) محاسبه می‌گردد.

$$\Delta PSNR = PSNR_{Proposed} - PSNR_{Ref}. \quad (5)$$

$$\Delta BR = \frac{BR_{Proposed} - BR_{Ref.}}{BR_{Ref.}} \times 100\% \quad (6)$$

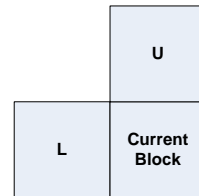
$$\Delta Time = \frac{T_{Propose} - T_{Ref.}}{T_{Ref.}} \times 100\% \quad (7)$$

جداول ۱ و ۲ نتایج شبیه‌سازی تصاویر QCIF و CIF را در QP‌های مختلف نشان می‌دهد. نتایج حاکی از آن است که

هر پیکسل در بلوک فعلی با Pi نشان داده می‌شود و m از رابطه (۴) محاسبه می‌گردد.

$$m = (\sum_{i=0}^{15} p_i + 8) >> 4 \quad (4)$$

مُد‌های محتمل دیگر می‌تواند از بلوک‌های چپ و بالایی بدست آید؛ شکل (۴) بلوک‌های مجاور بلوک فعلی را نشان می‌دهد [17].



شکل ۴- بلوک‌های مجاور بلوک فعلی [17]

با آزمون روی فریم‌های ویدیویی مختلف و با ساختار گوناگون مشاهده شد که احتمال آنکه (مُد فعلی = مُد U = مُد L) باشد ۴۰/۶۲٪ است و احتمال آنکه (مُد فعلی = مُد U) یا (مُد فعلی = مُد L) باشد ۸۰/۸۶٪ است. اما وقتی (مُد U = مُد L) است احتمال آنکه مُد فعلی برابر آنها باشد ۸۷/۵٪ است. بنابراین بررسی مُد‌های بلوک‌های مجاور U و L می‌تواند برای پیش‌تخمین بهترین مُد کاندید که با C نشانه‌گذاری شده استفاده گردد. لذا وقتی (مُد U = مُد L) است، تنها مُد کاندید C که برابر مُد بلوک‌های مجاور U و L می‌باشد بررسی می‌گردد؛ در این شرایط تعداد مُد‌های RD در بلوک‌های ۴×۴ از ۹ مُد، به یک مُد کاندید کاهش می‌یابد [17]. در غیر اینصورت الگوریتم انتخاب مُد داخلی پیشنهادی برای بلوک‌های روشنایی به‌صورت زیر خلاصه می‌شود:

مرحله ۱: برای بلوک ۴×۴ روشنایی، DDC از رابطه (۳) محاسبه گردد.

مرحله ۲-۱: اگر DDC بزرگتر از آستانه T1 (با مقدار ۳۲) است، روش RDO در نهایت برای ۴ مُد کاندید بررسی شود. دو مُد از کمترین مقدار محاسبه شده NSAD (C) بهترین مُد کاندید با کمترین مقدار NSAD و S با دومین کمترین مقدار NSAD به عنوان دومین مُد کاندید) و دو مُد از بلوک‌های مجاور و در صورتی که بلوک‌های مجاور شامل مُد DC باشند در محاسبه RDO مُد DC در نظر گرفته نمی‌شود (بدلیل آنکه DDC بزرگتر از آستانه T1 است) [18].

ماکزیمم زمان کدگذاری کاهش یافته به میزان ۲۹/۲۸٪ می- ناچیز است.
 باشد درحالی که افت کیفیت تصویر و افزایش نرخبیت بسیار

جدول ۱- نتایج شبیه‌سازی در تصاویر QCIF برای QPهای متفاوت

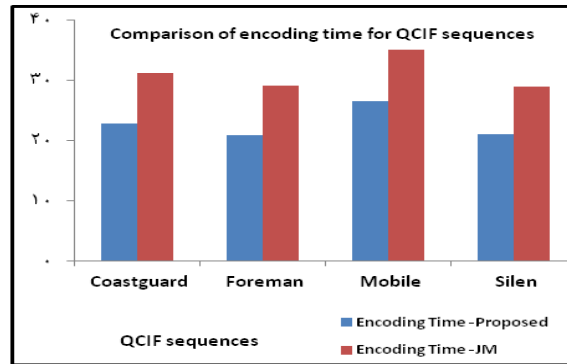
QP	Sequence	Δ PSNR(dB)	Δ BR(%)	Δ Time(%)
۱۶	Coastguard	-0.013	0.007	-23.7402
	foreman	-0.016	0.047	-28.7519
	Mobile	-0.021	0.004	-29.2775
	Silent	-0.016	0.0012	-26.3554
۲۸	Coastguard	-0.023	0.011	-27.0942
	foreman	-0.025	0.041	-28.0496
	Mobile	-0.015	0.016	-24.6228
	Silent	-0.01	0.005	-29.1798
۳۲	Coastguard	-0.031	0.028	-25.3247
	foreman	-0.035	0.046	-26.1614
	Mobile	-0.012	0.002	-28.0129
	Silent	-0.026	0.05	-29.5139
۴۴	Coastguard	0.046	0.035	-25.9205
	foreman	-0.014	0.0294	-22.1993
	Mobile	-0.023	0.037	-23.6253
	Silent	-0.038	0.025	-25.2765

جدول ۲- نتایج شبیه‌سازی در تصاویر CIF برای QPهای متفاوت

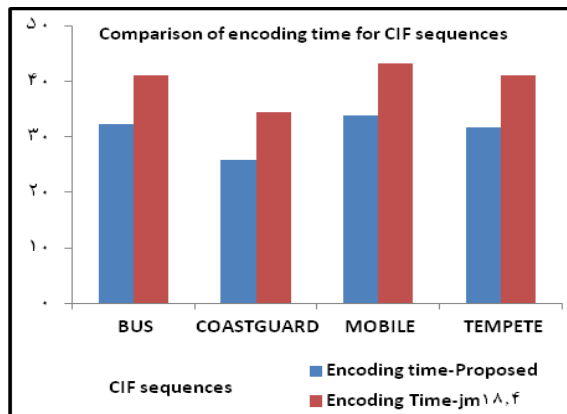
QP	Sequence	Δ PSNR(dB)	Δ BR(%)	Δ Time(%)
۱۶	Bus	-0.01	0.005	-21.2321
	Coastguard	-0.023	0.012	-21.4380
	Mobile	-0.013	0.017	-23.2457
	Tempete	-0.01	0.013	-23.9489
۲۸	Bus	-0.017	0.023	-21.2302
	Coastguard	-0.012	0.032	-24.6460
	Mobile	-0.015	0.072	-21.8864
	Tempete	-0.012	0.032	-22.7450
۳۲	Bus	-0.019	0.039	-25.9081
	Coastguard	-0.032	0.028	-22.5709
	Mobile	-0.02	0.027	-22.1182
	Tempete	-0.021	0.032	-28.4581
۴۴	Bus	-0.05	0.025	-27.4132

	Coastguard	-0.052	0.029	-23.9026
	Mobile	-0.044	0.032	-23.3868
	Tempete	-0.012	0.015	-22.5405

جدول ۳- مقایسه زمان کدگذاری روش پیشنهادی با نرم‌افزار مرجع در QP=28 در تصاویر نوع QCIF



جدول ۴- مقایسه زمان کدگذاری روش پیشنهادی با نرم‌افزار مرجع در QP=28 در تصاویر نوع CIF



جدول ۵- مقایسه PSNR

		Δ PSNR
		QP=32
FOREMAN	M1(Pan's Method [16])	-0.063
	M2 (Improved Pan's Method[16])	-0.001
	M3(new proposed Method[16])	-0.05
	Proposed Method	-0.035
SILENT	M1(Pan's Method [16])	-0.041
	M2 (Improved Pan's Method[16])	-0.023
	M3(new proposed Method[16])	-0.032
	Proposed Method	-0.026

پیچیدگی محاسباتی با تاثیر ناچیز در کیفیت تصویر و میزان PSNR بدست آمده است.

جداول ۸ و ۹ نتایج مقایسه با الگوریتم ارائه شده در [14] را در ویدیویی QCIF و CIF نشان می‌دهد. در روش پیشنهادی کاهش زمان کدگذاری با تاثیر ناچیز در میزان نرخ بیت و PSNR بدست آمده است.

جداول ۳ و ۴ مقایسه از زمان کدگذاری روش پیشنهادی و نرافزار مرجع JM18.4 را نشان می‌دهد. متوسط کاهش پیچیدگی کدگذار در تصاویر ویدیویی QCIF برابر ۲۲/۵٪ و در تصاویر از نوع CIF برابر ۲۶/۶٪ است. جداول ۵، ۶ و ۷ نتایج مقایسه با سایر الگوریتم‌های ارائه شده در [16] را نشان می‌دهد. در روش پیشنهادی کاهش

جدول ۶- مقایسه نرخ بیت

		$\Delta BR\%$
		QP=32
FOREMAN	M1(Pan's Method [16])	1.142
	M2 (Improved Pan's Method[16])	0.997
	M3(new proposed Method of[16])	0.887
	Proposed Method	0.046
SILENT	M1(Pan's Method [16])	0.905
	M2 (Improved Pan's Method[16])	0.590
	M3(new proposed Method of[16])	0.435
	Proposed Method	0.05

جدول ۷- مقایسه پیچیدگی

		$\Delta Time\%$
		QP=32
FOREMAN	M1(Pan's Method [16])	-49.10
	M2 (Improved Pan's Method[16])	-54.20
	M3(new proposed Method of[16])	-42.82
	Proposed Method	-26.1614
SILENT	M1(Pan's Method [16])	-45.38
	M2 (Improved Pan's Method[16])	-49.62
	M3(new proposed Method of[16])	-49.38
	Proposed Method	-29.5139

جدول ۸- مقایسه روش پیشنهادی و مرجع [14] در تصاویر QCIF

الگوریتم	Sequence		QP= 28	QP=32
[14]	coastguard	ΔT (%)	-62.73	-61
		$\Delta PSNR$ (dB)	-0.029	-0.03
		$\Delta Bit-rate$ (%)	0.59	0.94
روش پیشنهادی		ΔT (%)	-27.09	-25.3
		$\Delta PSNR$ (dB)	-0.023	-0.03
		$\Delta Bit-rate$ (%)	0.011	0.028

جدول ۹- مقایسه روش پیشنهادی و مرجع [14] در تصاویر CIF

الگوریتم	Sequence		QP= 28	QP=32
[14]	bus	ΔT (%)	-63.81	-62.3
		$\Delta PSNR$ (dB)	-0.063	-0.05
		$\Delta Bit-rate$ (%)	1.23	1.56
	mobile	ΔT (%)	-62.23	-61.5
		$\Delta PSNR$ (dB)	-0.065	-0.06
		$\Delta Bit-rate$ (%)	0.75	1
	tempete	ΔT (%)	-61.72	-60.5
		$\Delta PSNR$ (dB)	-0.065	-0.05
		$\Delta Bit-rate$ (%)	1.21	1.79
روش پیشنهادی	bus	ΔT (%)	-21.23	-25.9
		$\Delta PSNR$ (dB)	-0.017	-0.02
		$\Delta Bit-rate$ (%)	0.023	0.039
	mobile	ΔT (%)	-21.89	-22.1
		$\Delta PSNR$ (dB)	-0.015	-0.02
		$\Delta Bit-rate$ (%)	0.072	0.027
	tempete	ΔT (%)	-22.75	-28.5
		$\Delta PSNR$ (dB)	-0.012	-0.02
		$\Delta Bit-rate$ (%)	0.032	0.032

۵- نتیجه‌گیری و پیشنهادات

از نتایج شبیه‌سازی در بخش ۴ می‌توان نتیجه گرفت که الگوریتم پیشنهادی سریعتر از نرم‌افزار استاندارد JM18.4 است. به این دلیل که نرم‌افزار مرجع برای محاسبات RDO همه مدهای ممکن کدگذاری را بررسی کرده در حالی که با الگوریتم پیشنهادی تعداد مدهای ممکن جهت بررسی کاهش داده شد و با تشخیص سریع مدهای تخمین داخلی ۴×۴ در QP های پایین و ۱۶×۱۶ در QP های بالا بطور قابل ملاحظه‌ای سرعت کدگذاری افزایش یافت.

نتایج نشان داد که این روش می‌تواند به کاهش زمان محاسباتی تا ماکزیمم ۲۸/۲۹٪ با کاهش ناچیز در PSNR و افزایش نرخ بیت دست یابد. الگوریتم پیشنهادی در این مقاله ترکیب روشهای تصمیم‌د داخلی سریع روی بلوک‌های ۴×۴ و ۱۶×۱۶ با استفاده از پارامترهای کوتاه‌تیزه و استفاده از ماسک‌های جهت‌دار ساده و مدهای بلوک‌های همسایه در بلوک ۴×۴ می‌باشد که می‌تواند به بلوک‌های ۱۶×۱۶ نیز بسط داده شود. در این مقاله شبیه‌سازی بر روی تصاویر ویدیویی QCIF و CIF پیاده‌سازی شد و می‌تواند در سایر فرمت‌های ویدیویی چون HD نیز ادامه یابد.

منابع

1. Wiegand, T., Sullivan, G. J., Bjontegaard, G. J., Luthra, A. : 'Overview of the H.264/AVC Video Coding Standard', IEEE Trans. On circuits and systems for video technology, 13, (7), pp. 560-576, 2003.
2. Ghanbari, M.: 'Standard codecs: image compression to advanced video coding', 3rd edition, The Institution of Engineering and Technology, 2011.
3. Kim, C. S., Li, Q., Kuo, C. C. J.: 'Fast Intra-Prediction Model Selection for H.264 Codec', SPIE International Symposium ITCOM, Orlando, Florida, pp. 99-110, 2003.
4. Pan, F., Lin, X., Rahardja, S., Lim, K. P., Li, Z. G., Wu, D., Wu, S.: 'Fast Mode Decision Algorithm for Intra-prediction in H.264/AVC Video Coding', IEEE Trans. On circuits and systems for video Tech., 15, (7), pp. 813-822, 2005.
5. Pan, F., Lin, X., Rahardja, S., Lim, K. P., Li, Z. G., Wu, D., Wu, S.: 'Fast Mode Decision for Intra Prediction', JVT-G013, 7th Meeting: Pattaya II, Thailand, 7-14 March 2003.
6. Pan, F., Lin, X., Rahardja, S., Lim, K. P., Li, Z. G.: 'A Directional Field Based Fast Intra-Mode Decision Algorithm for H.264 Video Coding', IEEE Inter. Conf. on Multimedia and Expo, 2, pp. 1147-1150, June 2004.
7. Ren, Z., Ye, C., Liu, M.: 'A Fast Intra Prediction Algorithm for H.264', IEEE Computer Society, First International Workshop on Education Technology and Computer Science, 2,(2), pp. 772 - 775, March 2009.
8. Kim, J., Jeong, J.: 'Fast Intra-Mode Decision in H.264 Video Coding Using Simple Directional Masks', Proc. of SPIE, 5960, pp 1071-1079, July 2005.
9. Garg, R., Jindal, M., Chauhan, M.: 'Statistics Based Fast Intra- Mode Detection', Proc. of SPIE, 5960, pp 2085-2091, July 2005.
10. Jeon, B., Lee, J.: 'Fast Mode Decision for H.264', IEEE Inter. Conf. on Multimedia and Expo, 2, pp. 1131-1134, June 2004.
11. Hsu, C. L., Ho, M. H., Hong, J. J.: 'An Efficient Algorithm for Intra-Prediction Mode Selection in H.264', IEEE 7th International Conference, pp. 778-781, October 2007.
12. Elyousfi, A.: 'Gravity direction-based ultra-fast intraprediction algorithm for H.264/AVC video coding', Journal Signal, Image and Video Processing, 7,(1), pp. 53-65, 2013.
13. Wu, C. Y., Su, P. C. : 'Fast Intra-Coding for H.264/AVC by Using Projection-Based Predicted Block Residuals', IEEE Transactions on Multimedia , 15, (5), pp. 1083-1093, 2013.
14. Shang, J., Ding, W., Shi, Y., Sun, Y. : 'Fast Intra Mode Decision Algorithm Based on Texture Direction Detection for H.264/AVC', IJMECS,1, (5), pp.70-77, 2011.
15. Richardson, Iain EG: 'H.264 and MPEG-4 Video Compression', West Sussex: John Wiley & Sons, Ltd., ISBN 0-470-84837-5, 2003.
16. Jafari, M., Kasaei, S.: 'Fast Intra- and Inter-Prediction Mode Decision in H.264 Advanced Video Coding', International Journal of Computer Science and Network Security, 8, (5), pp. 130-140, 2008.
17. Pan, L. j., Ho, Y. S.: 'A Fast Mode Decision Algorithm for H.264/AVC Intra Prediction', IEEE Workshop in Signal Processing Systems, pp. 704-709, October 2007.
18. Muniyappa, S. k.: 'Implementation of Complexity Reduction Algorithm for Intra Mode Selection', Master of science Thesis, University of Texas, Arlington, December 2011.
19. Joint Video Team Reference Software JM 18.4:
<http://iphone.hhi.de/suehring/tml/download>.

