

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشکده برق و رباتیک

گروه الکترونیک

پایان نامه کارشناسی ارشد مهندسی برق-الکترونیک

ارائه روش جدیدی برای فشرده سازی تصاویر ویدئویی برای تلفن همراه با استفاده از
تخمین حرکت

مهسا اسماعیل پور قوچانی

استاد راهنما :

دکتر امیدرضا معروضی

استاد مشاور:

دکتر علیرضا احمدی فرد

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

ماه و سال انتشار: بهمن ۹۱

تقدیم به پدر بزرگوار و مادر مهربانم

آن دو فرشته ای که از خواسته هایشان گذشتند، سختی ها را به جان خریدند و خود را سپر برای مشکلات و ناملایمات کردند تا من به جایگاهی که اکنون در آن ایستاده ام برسم.

تقدیر و شکر

سپاس بی کران پروردگار یکتا را که، هستی مان بخشد و به طریق علم و دانش را بنمونمان شد و به، بنشیننی رهروان علم و دانش مفتخرمان نمود و خوشه چینی از علم و معرفت را روزیمان ساخت.

از پدر و مادر عزیزم که، همواره بر کوتاهی و درستی من، قلم عنو کشیده و گریانه از کنار غفلت بایم گذشته اند و در تمام عرصه های زندگی یار و یاور بی چشم داشت برای من بوده اند؛ از استاد با کمالات و شایسته جناب آقای دکتر امیر رضا معروضی که در کمال سه صدر، با حسن خلق و فروتنی، از هیچ کجی در این عرصه بر من دریغ ننمودند و زحمت راهنمایی این رساله را بر عهده گرفتند؛ بحال شکر و قدردانی را دارم.

چکیده

تخمین حرکت تطبیق بلوک در فشرده‌سازی تصاویر ویدئویی نقشی کلیدی دارد. تخمین حرکت هسته مرکزی در رمزگذاری ویدئو است، که حشویات زمانی میان فریم‌های ویدئویی مجاور را حذف کرده و راندمان رمزگذاری بالاتری را نتیجه می‌دهد.

در این پایان‌نامه دو الگوریتم تطبیق بلوک با هدف کاهش زمان جستجوی بلوک منطبق و بهبود کیفیت ویدئوی بازسازی شده پیشنهاد شده است. در الگوریتم اول از الگوی صلیبی و افقی با طول بازوهای نامساوی و موقعیت دو بردار حرکت همسایه بالا و چپ در مرحله جستجوی اولیه استفاده می‌شود. در الگوریتم دوم، در مرحله جستجوی اولیه سه الگوی مختلف پیشنهاد شده است؛ اول، الگوی مربعی بزرگ که اندازه گام‌های افقی و عمودی آن بر اساس بردار حرکت همسایه چپ تعیین می‌شود. دوم، الگوی بررسی چهار نقطه از الگوی مربعی بزرگ که در محدوده بردار حرکت همسایه چپ قرار دارند و سپس الگوی جستجو بر اساس مقادیر خطای بدست آمده از این نقاط تکمیل می‌شود. سوم، الگوی بررسی تنها چهار نقطه از الگوی مربعی بزرگ که در محدوده بردار حرکت همسایه چپ قرار دارند. برای هر دو الگوریتم در مرحله جستجوی محلی از الگوی مربعی 3×3 استفاده می‌شود.

نتایج حاصل از بررسی الگوریتم‌های پیشنهاد شده برای ۹ رشته تست ویدئو و مقایسه آن با الگوریتم‌های تطبیق بلوک استاندارد، نشان دهنده کاهش زمان جستجوی بلوک منطبق و افزایش سرعت پردازش بین ۲~۱۲ برابر نسبت به الگوریتم جستجوی کامل و بهبود PSNR به ویژه برای رشته‌های ویدئویی با جابجایی زیاد می‌باشد.

کلمات کلیدی: تخمین حرکت، رمزگذاری ویدئو، تطبیق بلوک، الگوی صلیبی و افقی، الگوی مربعی.

لیست مقالات مستخرج از پایان نامه

- ۱- تطبیق بلوک در رمزگذار ویدئو به روش پیش بینی متوسط بهبود یافته، یازدهمین کنفرانس سیستم‌های هوشمند ایران (ICIS 2013)
- 2- A novel and efficient motion estimation technique for video compression, 5th International Conference on Digital Image Processing (ICDIP 2013)

فهرست

عنوان	صفحه
۱ فصل اول.....	۲
۱-۱ مقدمه.....	۲
۲-۱ اهداف و کاربردها.....	۲
۳-۱ چالش‌های اصلی.....	۳
۱-۳-۱ راندمان رمزگذاری بالاتر	۳
۲-۳-۱ کاهش پیچیدگی محاسبات	۴
۳-۳-۱ بهبود انعطاف‌پذیری خطا	۴
۴-۱ راه‌حل‌های ممکن.....	۵
۵-۱ ساختار کلی پایان‌نامه.....	۷
۲ فصل دوم.....	۹
۱-۲ مقدمه.....	۹
۲-۲ روش‌های تفاضلی.....	۱۰
۳-۲ روش‌های پیکسل بازگشتی	۱۳
۴-۲ روش‌های حوزه فرکانس.....	۱۶
۵-۲ روش‌های تطبیق بلوک.....	۱۸
۱-۵-۲ تابع تطبیق	۲۰
۲-۵-۲ اندازه بلوک	۲۲
۳-۵-۲ محدوده جستجو	۲۳
۴-۵-۲ دقت جستجو	۲۴
۵-۵-۲ بردارهای حرکت نامحدود.....	۲۶
۶-۵-۲ ویژگی‌های ناحیه‌ها و سطوح خطای تطبیق بلوک.....	۲۷
۶-۲ مقایسه الگوریتم‌ها.....	۲۹
۳ فصل سوم.....	۳۷

۳۷	۱-۳ مقدمه.....
۳۸	۲-۳ الگوریتم جستجوی کامل (FS).....
۳۹	۳-۳ الگوریتم جستجوی سه مرحله‌ای (TSS).....
۴۰	۴-۳ الگوریتم جستجوی سه مرحله‌ای جدید (NTSS).....
۴۳	۵-۳ الگوریتم جستجوی چهار مرحله‌ای (FSS).....
۴۵	۶-۳ الگوریتم جستجوی لوزی (DS).....
۴۸	۷-۳ الگوریتم جستجوی الگوی صلیبی افقی (ARPS).....
۴۸	۱-۷-۳ نمایی از الگوریتم.....
۴۹	۲-۷-۳ پیشگویی بردار حرکت هدف.....
۵۰	۳-۷-۳ انتخاب الگوهای جستجو.....
۵۴	۴-۷-۳ پیش‌قضاوت حرکت صفر (ZMP).....
۵۵	۵-۷-۳ خلاصه روش ARPS-ZMP.....
۵۶	۸-۳ الگوریتم تطبیق بلوک به روش پیش‌بینی متوسط (MPBM).....
۶۰	۴ فصل چهارم.....
۶۰	۱-۴ مقدمه.....
۶۰	۲-۴ تطبیق بلوک به روش پیش‌بینی متوسط بهبود یافته.....
۶۵	۳-۴ الگوریتم جستجوی ساده و کارآمد مبتنی بر جستجوی سه مرحله‌ای.....
۷۴	۵ فصل پنجم.....
۷۴	۱-۵ مقدمه.....
۷۵	۲-۵ نتایج الگوریتم پیشنهادی اول.....
۷۵	۱-۲-۵ نتایج کمی.....
۸۹	۲-۲-۵ نتایج کیفی.....
۹۱	۳-۵ نتایج الگوریتم پیشنهادی دوم.....
۹۱	۱-۳-۵ نتایج کمی.....
۱۰۵	۲-۳-۵ نتایج کیفی.....
۱۰۸	۶ فصل ششم.....

۱-۶	مقدمه.....	۱۰۸
۲-۶	نتیجه گیری.....	۱۰۸
۳-۶	پیشنهادهات.....	۱۰۹
۷	مراجع.....	۱۱۱

فهرست شکل‌ها

- شکل ۱-۱: رمزگذار/رمزگشای ویدئو..... ۵
- شکل ۱-۲: تخمین حرکت تطبیق بلوک..... ۲۰
- شکل ۲-۲: کیفیت بازسازی برای SAD، SSD و NCCF..... ۲۲
- شکل ۳-۲: عملکرد BMA با اندازه‌های بلوک مختلف..... ۲۳
- شکل ۴-۲: عملکرد BMA با محدوده جستجوی مختلف..... ۲۴
- شکل ۵-۲: عملکرد BMA با دقت زیرپیکسل..... ۲۶
- شکل ۶-۲: عملکرد BMA با بردارهای حرکت محدود شده و نامحدود..... ۲۷
- شکل ۷-۲: توزیع متمرکز شده در مرکز ناحیه حرکتی بلوک..... ۲۸
- شکل ۸-۲: نمونه‌هایی از سطوح خطای چندوجهی..... ۲۹
- شکل ۹-۲: کیفیت پیشگویی الگوریتم‌های تخمین حرکت متفاوت..... ۳۳
- شکل ۱۰-۲: مقایسه بین الگوریتم‌های تخمین حرکت BMA و PRA..... ۳۴
- شکل ۱-۳: تخمین حرکت تطبیق بلوک..... ۳۷
- شکل ۲-۳: تخمین حرکت جستجوی کامل..... ۳۹
- شکل ۳-۳: الگوریتم جستجوی سه مرحله‌ای..... ۴۰
- شکل ۴-۳: الگوریتم جستجوی سه مرحله‌ای جدید..... ۴۲
- شکل ۵-۳: الگوی جستجوی FSS..... ۴۴
- شکل ۶-۳: پروسه الگوریتم FSS..... ۴۵
- شکل ۷-۳: (الف) الگوی جستجوی لوزی بزرگ (LDSP)، (ب) الگوی جستجوی لوزی کوچک (SDSP)..... ۴۶
- شکل ۸-۳: مثالی از مسیر جستجوی الگوریتم DS..... ۴۷
- شکل ۹-۳: چهار نوع ROS مشخص شده با بلوک‌های تیره..... ۵۰
- شکل ۱۰-۳: الگوی صلیبی افقی (ARP)..... ۵۱
- شکل ۱۱-۳: دو الگوی جستجوی ثابت..... ۵۴
- شکل ۱۲-۳: الگوی جستجو در مرحله جستجوی اولیه..... ۵۸
- شکل ۱-۴: موقعیت ماکروبلوک‌های پیشگویی..... ۶۱
- شکل ۲-۴: الگوهای جستجو در مراحل جستجوی اولیه و محلی..... ۶۳

- شکل ۴-۳: موقعیت ماکروبلوک پیشگویی..... ۶۶
- شکل ۴-۴: الگوهای جستجو در مراحل جستجوی اولیه و محلی،..... ۶۷
- شکل ۴-۵: جهت‌های جستجو در الگوی جستجو..... ۶۹
- شکل ۴-۶: چهار ناحیه جستجو..... ۷۰
- شکل ۴-۷: الگوهای جستجو بر اساس هر یک از یک چهارم‌های انتخاب شده..... ۷۱
- شکل ۴-۸: نمونه‌ای از موقعیت نقاط جستجو در مرحله جستجوی اولیه..... ۷۲
- شکل ۵-۱: مقایسه عملکرد بر حسب MSE روی فریم برای سرکارگر..... ۸۱
- شکل ۵-۲: مقایسه عملکرد بر حسب MSE روی فریم برای تلفن ماشین..... ۸۱
- شکل ۵-۳: مقایسه عملکرد بر حسب MSE روی فریم برای فوتبال..... ۸۲
- شکل ۵-۴: مقایسه عملکرد بر حسب MSE روی فریم برای اتوبوس..... ۸۲
- شکل ۵-۵: مقایسه عملکرد بر حسب Δ MSE روی فریم برای سرکارگر..... ۸۳
- شکل ۵-۶: مقایسه عملکرد بر حسب Δ MSE روی فریم برای تلفن ماشین..... ۸۴
- شکل ۵-۷: مقایسه عملکرد بر حسب Δ MSE روی فریم برای فوتبال..... ۸۴
- شکل ۵-۸: مقایسه عملکرد بر حسب Δ MSE روی فریم برای اتوبوس..... ۸۵
- شکل ۵-۹: مقایسه عملکرد بر حسب PSNR روی فریم برای سرکارگر..... ۸۵
- شکل ۵-۱۰: مقایسه عملکرد بر حسب PSNR روی فریم برای تلفن ماشین..... ۸۶
- شکل ۵-۱۱: مقایسه عملکرد بر حسب میانگین PSNR برای فوتبال..... ۸۶
- شکل ۵-۱۲: مقایسه عملکرد بر حسب میانگین PSNR برای اتوبوس..... ۸۷
- شکل ۵-۱۳: مقایسه عملکرد بر حسب Δ PSNR روی فریم برای سرکارگر..... ۸۷
- شکل ۵-۱۴: مقایسه عملکرد بر حسب Δ PSNR روی فریم برای تلفن ماشین..... ۸۸
- شکل ۵-۱۵: مقایسه عملکرد بر حسب Δ PSNR روی فریم برای فوتبال..... ۸۸
- شکل ۵-۱۶: مقایسه عملکرد بر حسب Δ PSNR روی فریم برای اتوبوس..... ۸۹
- شکل ۵-۱۷: تصویر بازسازی شده فریم شماره ۱۴ رشته سرکارگر..... ۹۰
- شکل ۵-۱۸: مقایسه عملکرد بر حسب MSE روی فریم برای سرکارگر..... ۹۷
- شکل ۵-۱۹: مقایسه عملکرد بر حسب MSE روی فریم برای تلفن ماشین..... ۹۷
- شکل ۵-۲۰: مقایسه عملکرد بر حسب MSE روی فریم برای فوتبال..... ۹۸

- شکل ۵-۲۱: مقایسه عملکرد بر حسب MSE روی فریم برای اتوبوس..... ۹۸
- شکل ۵-۲۲: مقایسه عملکرد بر حسب Δ MSE روی فریم برای سرکارگر..... ۹۹
- شکل ۵-۲۳: مقایسه عملکرد بر حسب Δ MSE روی فریم برای تلفن ماشین..... ۹۹
- شکل ۵-۲۴: مقایسه عملکرد بر حسب Δ MSE روی فریم برای فوتبال..... ۱۰۰
- شکل ۵-۲۵: مقایسه عملکرد بر حسب Δ MSE روی فریم برای اتوبوس..... ۱۰۰
- شکل ۵-۲۶: مقایسه عملکرد بر حسب میانگین PSNR برای سرکارگر..... ۱۰۱
- شکل ۵-۲۷: مقایسه عملکرد بر حسب میانگین PSNR برای تلفن ماشین..... ۱۰۱
- شکل ۵-۲۸: مقایسه عملکرد بر حسب میانگین PSNR برای فوتبال..... ۱۰۲
- شکل ۵-۲۹: مقایسه عملکرد بر حسب میانگین PSNR برای اتوبوس..... ۱۰۲
- شکل ۵-۳۰: مقایسه عملکرد بر حسب Δ PSNR روی فریم برای سرکارگر..... ۱۰۳
- شکل ۵-۳۱: مقایسه عملکرد بر حسب Δ PSNR روی فریم برای تلفن ماشین..... ۱۰۳
- شکل ۵-۳۲: مقایسه عملکرد بر حسب Δ PSNR روی فریم برای فوتبال..... ۱۰۴
- شکل ۵-۳۳: مقایسه عملکرد بر حسب Δ PSNR روی فریم برای اتوبوس..... ۱۰۴
- شکل ۵-۳۴: تصویر بازسازی شده فریم شماره ۱۹ رشته تلفن ماشین..... ۱۰۶

فهرست جداول

جدول ۱-۲: پیچیدگی محاسباتی SAD، SSD و NCCF برای بلوک $N \times N$	۲۲
جدول ۱-۵: میانگین MSE برای ۳۰ فریم اول رشته‌های آزمون.....	۷۶
جدول ۲-۵: میانگین PSNR برای ۳۰ فریم اول رشته‌های آزمون.....	۷۷
جدول ۳-۵: زمان مورد نیاز برای پردازش ۳۰ فریم اول رشته‌های آزمون بر حسب ثانیه.....	۷۸
جدول ۴-۵: سرعت پردازش.....	۷۹
جدول ۵-۵: میانگین MSE برای ۳۰ فریم اول رشته‌های آزمون.....	۹۲
جدول ۶-۵: میانگین PSNR برای ۳۰ فریم اول رشته‌های آزمون.....	۹۳
جدول ۷-۵: زمان مورد نیاز برای پردازش ۳۰ فریم اول رشته‌های آزمون بر حسب ثانیه.....	۹۴
جدول ۸-۵: سرعت پردازش.....	۹۵

فهرست اختصارات

ARPS	Adaptive Rood Pattern Search
BDM	Block Distortion Measure
BMA	Block-Matching Algorithm
BMME	Block-Matching Motion Estimation
CIF	Common Intermediate Format
DF	Displaced Frame
DFA	Differential Frame Algorithhm
DFD	Displaced-Frame Difference
DS	Diamond Search
EMPBM	Enhanced Mean Predictive Block Matching
FS	Full Search
FSS	Four Step Search
GSM	Global System for Mobile
LDSP	Large Diamond Search Pattern
MC	Motion Compensation
MCP	Motion-Compensated Prediction
ME	Motion Estimation
MME	Minimal Matching Error
MPBM	Mean Predictive Block Matching
MV	Motion Vector

NCCF	Normalized Cross-Correlation Function
NTSS	New Three Step Search
PCA	Phase Correlation Algorithm
PRA	Pixel-Recursive Algorithm
PSNR	Peak Signal to Noise Ratio
ROS	Region Of Support
SAD	Sum of Absolute Difference
SDSP	Small Diamond Search Pattern
SSD	Sum of Squared Difference
TSS	Three Step Search
URP	Unit-size Rood Pattern
VLC	Variable-Length Coding
ZMP	Zero-Motion Prejudgment

فصل اول

مقدمه

۱ فصل اول

۱-۱ مقدمه

در ابتدای این فصل اهداف و انگیزه‌های انجام این تحقیق آورده شده و برخی از کاربردهای آن مطرح می‌گردد. سپس چالش‌های موجود در زمینه مخابرات تلفن همراه بیان شده و راه‌حل‌های ممکن ارائه می‌شود. در انتها ساختار کلی پایان‌نامه توضیح داده می‌شود.

۲-۱ اهداف و کاربردها

در سال‌های اخیر، دو تکنولوژی مجزای مخابرات چند رسانه‌ای^۱ و مخابرات تلفن همراه^۲ بسیار رشد کرده و موفقیت تجاری بزرگی بدست آوردند. با افزایش تحرک اشخاص و افزایش تکیه بر قابلیت در دسترس بودن اطلاعات چند رسانه‌ای، نیاز بزرگی برای بهبود اطلاعات چند رسانه‌ای روی حرکت احساس شد. انگیزه موجود در این دیدگاه توانایی مخابره هر نوع اطلاعات از هر مکان و در هر زمان می‌باشد، که همگرایی مخابرات چند رسانه‌ای و تلفن همراه را موجب می‌شود. این حوزه جدید، مخابرات چند رسانه‌ای تلفن همراه^۳ نامیده می‌شود. عبارت چند رسانه‌ای به نمایش اطلاعات اشکال مختلف رسانه اشاره دارد، که شامل رسانه متنی، رسانه عددی، رسانه شنیداری، رسانه دیداری و هر شکل دیگر نمایش اطلاعات می‌باشد.

سیستم‌های مخابراتی تلفن همراه نسل دوم فعلی، مانند سیستم جهانی برای تلفن همراه^۴ (GSM)، تعدادی از سرویس‌های مخابراتی چند رسانه‌ای پایه مانند صدا، سرویس‌های پیام کوتاه، نامه الکترونیکی، مخابره تصویر ثابت، و دستیابی به اینترنت پایه را پشتیبانی

^۱Multimedia communications

^۲Mobile communications

^۳Mobile multimedia communications

^۴Global System for Mobile

می‌کنند. با این حال، برای دستیابی به مخابره بلادرنگ ویدئو در تلفن همراه ابتدا باید بسیاری از مسائل فناوری مورد تقاضا حل شود. هنگامی که این مسائل حل شوند، کاربردها و سرویس‌های پیشرفته بسیاری برای کاربران تلفن همراه در دسترس خواهد بود؛ از جمله:

- آموزش و یادگیری از راه دور
- بازی متقابل
- خرید از راه دور
- سرویس‌های رسانه‌ای برخط، مانند گزارش‌های خبری
- ویدئو تلفن
- ویدئو کنفرانس

۱-۳ چالش‌های اصلی

۱-۳-۱ راندمان رمزگذاری بالاتر

طیف رادیویی بسیار محدود و منابع کمیاب هستند. از اینرو محدودیت‌های بسیار زیادی را روی پهنای باند موجود برای کانال تلفن همراه بوجود می‌آورند. با توجه به میزان مشخص دیتای تولید شده توسط ویدئو، استفاده از فنون رمزگذاری کارآمد ضروری است. برای مثال، ارسال بلادرنگ ویدئو با قالب میانی مشترک^۱ (CIF) [۱] با ۱۵ فریم بر ثانیه روی کانال GSM ۹/۶ کیلوبیت بر ثانیه، به نرخ فشرده‌سازی ۱:۱۹۰۰ نیاز دارد. اگرچه فنون رمزگذاری فعلی قابلیت ایجاد چنین آهنگ فشرده‌سازی را دارند، اما به راندمان رمزگذاری بالاتری به منظور

^۱ Common Intermediate Format

بهبود کیفیت (یعنی قالب‌های^۱ بزرگتر، آهنگ فریم بالاتر، و کیفیت بصری بهتر) ویدئو در آهنگ بیت بسیار کم نیاز است.

۱-۳-۲ کاهش پیچیدگی محاسبات

در پایانه‌های تلفن همراه، توان پردازش و عمر باتری بسیار محدود است. با توجه به میزان قابل توجه توان محاسباتی مورد نیاز برای پردازش ویدئو، استفاده از فنون کاهش دهنده پیچیدگی ضروری است.

۱-۳-۳ بهبود انعطاف‌پذیری خطا

کانال تلفن همراه به دلیل تعدادی از مکانیسم‌های تلفاتی از قبیل محوشدگی^۲، سایه اندازی^۳، و تداخل بین کانالی^۴ محیطی با آهنگ خطای بیت بالا می‌باشد. در ویدئو، اثرات چنین خطاهایی تقویت می‌شوند زیرا رشته بیت در ویدئو به دلیل محدودیت‌های پهنای باند بسیار فشرده می‌شود. در حقیقت، فشرده سازی بالاتر حساسیت رشته بیت به خطا را بیشتر می‌کند، زیرا در این حالت هر بیت نماینده مقدار بیشتری از ویدئوی رمزگشایی شده می‌باشد. همچنین اثرات خطا روی ویدئو به دلیل استفاده از رمزگذاری پیشگویانه^۵ و رمزگذاری با طول متغیر^۶ (VLC) تقویت می‌شود. استفاده از چنین روش‌های رمزگذاری می‌تواند منتهی به انتشار خطای زمانی و فضایی شود. بنابراین با ارسال روی کانال تلفن همراه، ویدئوی فشرده شده افت کیفی شدیدی را متحمل می‌شود و استفاده از فنون انعطاف خطا^۷ ضروری است.

^۱ Format

^۲ Multipath fading

^۳ Shadowing

^۴ Co-channel interference

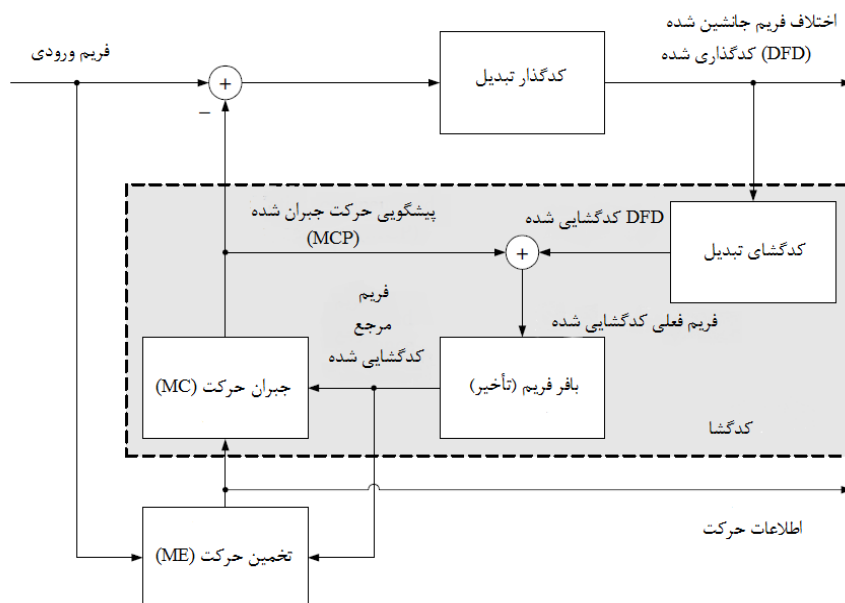
^۵ Predictive coding

^۶ Variable-Length Coding

^۷ Error-resilience

۴-۱ راه حل های ممکن

مخابره ویدئو در تلفن همراه یک موضوع میان رشته‌ای می‌باشد [۲]. خوشبختانه سیستم‌های کامل راه‌حل‌هایی را در بخش‌های مختلف نظیر رمزگذاری منبع ویدئو، رمزگذاری کانال، طراحی شبکه، طراحی نیمه‌هادی، و غیره اعمال کرده‌اند. در این پایان‌نامه تمرکز روی راه‌حل مبتنی بر رمزگذاری منبع ویدئو می‌باشد. بنابراین، ابتدا باید بطور دقیق رمزگذاری منبع ویدئو مورد بررسی قرار گیرد. شکل ۱-۱ رمزگذار/رمزگشای ویدئو را نشان می‌دهد.



شکل ۱-۱: رمزگذار/رمزگشای ویدئو [۱]

اساساً تغییرات بین فریم‌های متوالی در رشته ویدئو به دلیل حرکت اشیاء می‌باشد. در نتیجه، بلوک تخمین حرکت^۱ (ME) از یک مدل حرکتی برای تخمین جابجایی بین فریم فعلی و فریم مرجع (معمولاً فریم رمزگشایی شده قبلی در بافر فریم ذخیره می‌شود) استفاده

^۱Motion Estimation

می‌کند. سپس این اطلاعات حرکت توسط بلوک جبران حرکت^۱ (MC) مورد استفاده قرار می‌گیرد. این بلوک محتوای فریم مرجع را به منظور ارائه پیشگویی از فریم فعلی حرکت می‌دهد. پیشگویی حرکت جبران شده^۲ (MCP)، فریم جانشین شده^۳ (DF) نیز نامیده می‌شود. سپس فریم پیشگویی شده از فریم فعلی تفریق می‌شود تا سیگنال خطا به نام اختلاف فریم جانشین شده^۴ (DFD) بدست آید. به جای رمزگذاری فریم فعلی، این سیگنال خطا رمزگذاری می‌شود، زیرا آنتروپی آن کمتر است. در رمزگشا، همان فریم مرجع به همراه اطلاعات حرکت دریافت شده برای ایجاد همان پیشگویی مورد استفاده قرار می‌گیرد. سپس این پیشگویی به سیگنال خطای دریافت شده اضافه شده و فریم فعلی بازسازی می‌شود.

بررسی دقیق این رمزگذار/رمزگشا نشان می‌دهد که روش مبتنی بر حرکت می‌تواند راه‌حل‌های مناسبی را برای سه چالش راندمان رمزگذاری بالاتر، کاهش پیچیدگی، و انعطاف پذیری خطا ارائه دهد. این روش مبتنی بر حرکت بصورت زیر خلاصه می‌شود:

۱. فنون تخمین حرکت پیشرفته. یک روش برای دستیابی به راندمان رمزگذاری

بالاتر، بهبود عملکرد فرآیندهای تخمین و جبران حرکت می‌باشد. هدف بدست آوردن پیشگویی حرکت جبران شده بهتر و در نتیجه کاهش آنتروپی سیگنال DFD می‌باشد.

۲. فنون تخمین حرکت کاهش پیچیدگی. در رمزگذار/رمزگشای ویدئو تخمین

حرکت یک فرآیند بسیار محاسباتی است. در حقیقت، پیچیدگی محاسباتی این فرآیند بیشتر از مجموع سایر مراحل رمزگذاری می‌باشد. بنابراین با کاهش محاسبات این فرآیند، پیچیدگی کلی رمزگذار/رمزگشا کاهش می‌یابد.

^۱Motion Compensation

^۲Motion-Compensated Prediction

^۳Displaced Frame

^۴Displaced-Frame Difference

۳. فنون پوشاندن خطای حرکت جبران شده. صرفنظر از دیتای کنترلی و سرپیام،

خروجی رمزگذار/رمزگشای ویدئو یکی از دو نوع زیر است: دیتای حرکت^۱ و دیتای

خطا (یعنی DFD). در میان این دو نوع، معمولاً دیتای حرکت بیشتر اطلاعات فریم

را حمل می‌کند. در حقیقت، در آهنگ بیت کم (مانند مخابره ویدئو در تلفن همراه)،

دیتای حرکت درصد بسیار زیادی از بیت‌های در دسترس را مصرف می‌کند [۳].

بنابراین زمانی که خطا رخ می‌دهد بازیابی اطلاعات حرکت از دست رفته یا دریافت

شده بصورت اشتباه، بسیار مهم می‌باشد. یک کلاس فنون انعطاف خطا که به ویژه

برای مخابرات ویدئویی تلفن همراه مناسب می‌باشد، پوشاندن خطای حرکت جبران

شده یا پوشاندن خطای زمانی^۲ است، زیرا برخلاف دیگر تکنیک‌ها، نرخ بیت را

افزایش نمی‌دهد و هیچ تأخیری را ایجاد نمی‌کند.

۱-۵ ساختار کلی پایان‌نامه

در فصل اول این پایان‌نامه، مقدمه‌ای در رابطه با مخابرات تلفن همراه بیان شده است. در

فصل دوم مروری بر روش‌های تخمین حرکت ارائه می‌شود، که در واحد رمزگذار/رمزگشای

ویدئو بکار می‌رود. در فصل سوم تعدادی از الگوریتم‌های تطبیق بلوک، که یکی از روش‌های

تخمین حرکت می‌باشد، توضیح داده می‌شود. در فصل چهارم دو الگوریتم تطبیق بلوک جدید

پیشنهاد شده است و در فصل پنجم نتایج الگوریتم‌های پیشنهادی مورد بررسی قرار می‌گیرد.

در فصل ششم نتیجه‌گیری و کار آتی بیان می‌شود.

^۱Motion data

^۲Temporal error concealment

فصل دوم

مروری بر روش های تخمین حرکت

۲ فصل دوم

۲-۱ مقدمه

روش معمول برای رمزگذاری ویدئو، رمزگذاری حرکت جبران شده می باشد. در اولین مرحله این روش، که تخمین حرکت (ME) نامیده می شود، حرکت اشیا بین فریم مرجع و فریم فعلی تخمین زده می شود. سپس این اطلاعات حرکت در مرحله دوم، که جبران حرکت (MC) نامیده می شود، استفاده می شوند تا اشیا در فریم مرجع را حرکت داده و یک پیشگویی برای فریم فعلی ارائه شود. به جای فریم فعلی خطای پیشگویی، که اختلاف فریم جانشین شده (DFD) نامیده می شود، رمزگذاری می گردد. اطلاعات حرکت تخمین زده شده نیز باید ارسال شود، مگر اینکه رمزگشا بتواند آن را از اطلاعات رمزگشایی شده قبلی تخمین بزند.

فریم مرجع بکار رفته در ME، از نظر زمانی قبل یا بعد از فریم قبلی قرار دارد. این دو حالت به ترتیب پیشگویی پیش رو^۱ و پیشگویی پس رو^۲ نامیده می شوند. در پیشگویی دوجهته^۳، از دو فریم مرجع (یکی برای پیشگویی پیش رو و دیگری برای پیشگویی پس رو) استفاده می شود و پیشگویی ها با یکدیگر تلفیق می شوند، که فریم پیشگویی شده بدست آمده B-فریم نامیده می شود.

تخمین حرکت یک فرآیند مهم در محدوده وسیعی از رشته های علمی و کاربردها از قبیل تحلیل رشته تصویر، بینایی کامپیوتر، ردیابی هدف، و رمزگذاری ویدئو است. رشته های

^۱Forward prediction

^۲Backward prediction

^۳Bidirectional prediction

علمی و کاربردهای مختلف نیازمندی های مختلفی دارند و در نتیجه از فنون تخمین حرکت مختلفی استفاده می کنند.

در ادامه این فصل تعدادی از فنون تخمین حرکت بررسی می شود و در انتها مقایسه ای بین فنون توضیح داده شده صورت می گیرد.

۲-۲ روش های تفاضلی

روش های تفاضلی^۱ براساس رابطه بین تغییرات فضایی^۲ و زمانی^۳ شدت روشنایی عمل می کنند. این روش ها ابتدا توسط لیمب^۴ و مرفی^۵ در سال ۱۹۷۵ پیشنهاد شد [۴]. در روش پیشنهادی، آن ها از دامنه اختلاف فریم زمانی (FD) روی ناحیه حرکت A استفاده کردند، تا سرعت این ناحیه را اندازه گیری نمایند. به منظور حذف وابستگی به اندازه ناحیه، این اندازه گیری توسط اختلاف پیکسل فضایی افقی (HD)، یا عمودی (VD) نرمالیزه می شود. در نتیجه بردار حرکت تخمین زده شده بصورت زیر بدست می آید:

$$\hat{d} = \begin{bmatrix} \hat{d}_x \\ \hat{d}_y \end{bmatrix} = \begin{bmatrix} \frac{\sum_{s \in A} FD(s) \text{sign}(HD(s))}{\sum_{s \in A} |HD(s)|} \\ \frac{\sum_{s \in A} FD(s) \text{sign}(VD(s))}{\sum_{s \in A} |VD(s)|} \end{bmatrix} \quad (1-2)$$

که

$$\text{sign}(z) = \begin{cases} \frac{z}{|z|}, & \text{if } |z| \geq \text{threshold}, \\ 0, & \text{otherwise,} \end{cases} \quad (2-2)$$

$$FD(s) = f_t(s) - f_{t-\Delta t}(s), \quad (3-2)$$

$$HD(s) = \frac{1}{2} [f_t(x+1, y) - f_t(x-1, y)], \quad (4-2)$$

^۱Differential method

^۲Spatial

^۳Temporal

^۴Limb

^۵Murphy

$$VD(s) = \frac{1}{2} [f_t(x, y + 1) - f_t(x, y - 1)]. \quad (5-2)$$

قبلی از نظر زمانی در موقعیت پیکسل (x, y) و در زمان t و $f_{t-\Delta t}(x, y)$ فریم مرجع $f_t(x, y)$ فریم فعلی در موقعیت پیکسل (x, y) و در زمان $t-\Delta t$ را نشان می دهد.

بعد از آن، پایه نظری روش های تفاضلی توسط کافوریو^۱ و رکا^۲ در سال ۱۹۷۶ بنا نهاده شد [۵]. آن ها با تعریف پایه اختلاف فریم، معادله (۲-۳)، شروع کرده و آن را بصورت زیر نوشتند:

$$\begin{aligned} FD(s) &= f_t(s) - f_{t-\Delta t}(s) \\ &= f_t(s) - f_t(s + d) \end{aligned} \quad (6-2)$$

مقادیر کوچک d ، سمت راست معادله (۲-۶)، می تواند با بسط سری تیلور حول s بصورت زیر جایگزین شود:

$$FD(s) = -d^T \nabla_s f_t(s) + \text{عبارات مرتبه بالاتر} \quad (7-2)$$

که $\nabla_s = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right]^T$ گرادیان فضایی با توجه به s می باشد. صرفنظر از عبارات مرتبه بالاتر و این فرض که حرکت روی ناحیه A ثابت است، از رگرسیون خطی برای بدست آوردن تخمین حداقل متوسط خطای d می توان استفاده کرد.

$$\hat{d} = -[\sum_{s \in A} \nabla_s f_t(s) \nabla_s^T f_t(s)]^{-1} [\sum_{s \in A} FD(s) \nabla_s f_t(s)] \quad (8-2)$$

این معادله وابستگی زیادی به گرادیان فضایی ∇_s دارد. به همین دلیل، روش های تفاضلی را روش های گرادیان^۳ نیز می نامند.

^۱ Cafforio

^۲ Rocca

^۳ Gradient methods

با استفاده از تقریب $\nabla_s f_t(s) \approx [HD(s), VD(s)]^T$ معادله (۸-۲) بصورت زیر خلاصه می شود:

$$\hat{a} = - \begin{bmatrix} \sum_{s \in A} HD^2(s) & \sum_{s \in A} HD(s) \cdot VD(s) \\ \sum_{s \in A} HD(s) \cdot VD(s) & \sum_{s \in A} VD^2(s) \end{bmatrix}^{-1} \times \begin{bmatrix} \sum_{s \in A} FD(s) \cdot HD(s) \\ \sum_{s \in A} FD(s) \cdot VD(s) \end{bmatrix} \quad (9-2)$$

با صرف نظر از عبارات متقاطع (یعنی $\sum_{s \in A} HD(s) \cdot VD(s) \approx 0$) می توان نشان داد که راه حل تحلیلی کافوریو و رکا (معادله (۹-۲)) به سادگی راه حل ابتکاری لیمب و مرفی (معادله (۱-۲)) می شود.

نقص اصلی روش های تفاضلی این است که آن ها می توانند تنها برای اندازه گیری جابجایی های حرکتی کوچک (تا حدود ± 3 پیکسل) مورد استفاده قرار بگیرند. برای غلبه بر این مشکل تعدادی راه حل مانند روش تکراری یاماگوچی^۱ [۶] پیشنهاد شده است. در این روش، ابتدا یک بردار حرکت اولیه با استفاده از معادله (۹-۲) بین بلوک در فریم فعلی و بلوک مورد نظر در موقعیت مشابه در فریم مرجع تخمین زده می شود. در تکرار بعد، موقعیت بلوک منطبق در فریم مرجع توسط بردار حرکت اولیه جابجا می شود، و سپس روش تفاضلی دوباره بکار برده می شود تا تخمین دوم بدست آید. این تخمین دوم بعنوان یک عبارت اصلاحی برای تخمین اولیه عمل می کند. این فرآیند جابجایی و تخمین ادامه پیدا می کند تا زمانی که عبارت اصلاحی به اندازه کافی کوچک شود.

نقص دیگر روش های تفاضلی این است که عملگر گرادیان فضایی، ∇_s ، به نویز دیتا حساس است. این نقص می تواند با استفاده از مجموعه دیتای بزرگتر در محاسبات کمتر شود.

^۱Yamaguchi

مواردی وجود دارد که در آن‌ها روش‌های تفاضلی بدرستی عمل نمی‌کنند [۷]. برای مثال، در نواحی هموار گرادیان تقریباً برابر صفر است و ماتریس معادله (۲-۹) منفرد^۱ می‌شود. هم‌چنین، هنگامیکه حرکت با لبه‌ها در تصویر موازی است، یعنی $d^T \nabla_s \approx 0$ ، اختلاف فریم، معادله (۲-۷)، صفر می‌شود و جابجایی اشتباه صفر را نتیجه می‌دهد. چنین مشکلاتی ممکن است با افزایش ناحیه دیتا تا حدی حل شود، اما می‌تواند مشکل دقت را افزایش دهد.

۲-۳ روش‌های پیکسل بازگشتی

با توجه به تابع چند مجهولی $g(\mathbf{r})$ ، $\mathbf{r}=[r_1, \dots, r_n]^T$ ، اغلب روش مستقیم برای حداقل کردن آن، محاسبه مشتقات جزئی آن تابع نسبت به هر مجهول و برابر قرار دادن آن‌ها با صفر و حل معادلات همزمانی بدست آمده می‌باشد. این روش، بهینه‌سازی مبتنی بر گرادیان^۲ نامیده می‌شود و نمایش برداری آن بصورت زیر است:

$$\nabla_{\mathbf{r}} g(\mathbf{r}) = 0 \quad (۲-۱۰)$$

در مواردی که تابع $g(\mathbf{r})$ نمی‌تواند به فرم بسته نمایش داده شود و یا مجموعه معادلات همزمانی (۲-۱۰) حل نمی‌شود، روش‌های تکرار کننده عددی^۳ بکار برده می‌شود.

یکی از ساده‌ترین روش‌های عددی روش نزول سریع^۴ می‌باشد.

^۱ Singular

^۲ Gradient-based optimization

^۳ Numerical iterative methods

^۴ Steepest-descent method

از آنجایی که بردار گرادیان جهت حداکثر را نشان می‌دهد، این روش تخمین فعلی موقعیت حداقل در جهت منفی گرادیان، $\hat{\mathbf{r}}^i$ ، را به روز می‌کند تا تخمین بهبود یافته جدیدی بدست آید:

$$\hat{\mathbf{r}}^{i+1} = \hat{\mathbf{r}}^i - \alpha \nabla_{\mathbf{r}} g(\hat{\mathbf{r}}^i) \quad (11-2)$$

که $\alpha > 0$ اندازه گام به روز رسانی و i شاخص تکرار می‌باشد.

روش‌های پیکسل بازگشتی^۱ بر اساس تکرار حداقل سازی مبتنی بر گرادیان خطای پیشگویی عمل می‌کنند. این روش‌ها ابتدا توسط نتراولی^۲ و رابینز^۳ در ۱۹۷۹ پیشنهاد شد [۸]. در الگوریتم پیشنهادی، آن‌ها از روش نزول سریع استفاده می‌کنند تا مجذور اختلاف فریم جانشین شده، $DFD(\mathbf{s}, \mathbf{d})$ ، را با توجه به بردار جابجایی، $\mathbf{d}$ ، بصورت تکرار شونده حداقل کنند. بنابراین

$$g(\mathbf{r}) = DFD^2(\mathbf{s}, \mathbf{d}) \quad (12-2)$$

که

$$DFD(\mathbf{s}, \mathbf{d}) = f_t(\mathbf{s}) - f_{t-\Delta t}(\mathbf{s} - \mathbf{d}). \quad (13-2)$$

با جانشین کردن معادله (۱۲-۲) در معادله (۱۱-۲) و قرار دادن $\alpha = \frac{\epsilon}{2}$ داریم:

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^i - \frac{\epsilon}{2} \nabla_{\mathbf{d}} DFD^2(\mathbf{s}, \hat{\mathbf{d}}^i). \quad (14-2)$$

اکنون،

$$\nabla_{\mathbf{d}} DFD^2(\mathbf{s}, \mathbf{d}) = 2DFD(\mathbf{s}, \mathbf{d}) \nabla_{\mathbf{d}} DFD(\mathbf{s}, \mathbf{d})$$

^۱ Pel-recursive methods

^۲ Netravali

^۳ Robbins

$$\begin{aligned}
&= 2\text{DFD}(\mathbf{s}, \mathbf{d}) \nabla_{\mathbf{d}} [f_t(\mathbf{s}) - f_{t-\Delta t}(\mathbf{s} - \mathbf{d})] \\
&= 2\text{DFD}(\mathbf{s}, \mathbf{d}) \nabla_{\mathbf{s}} f_{t-\Delta t}(\mathbf{s} - \mathbf{d}) \quad (15-2)
\end{aligned}$$

با جایگزینی معادله (15-2) در معادله (14-2) داریم

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^i - \varepsilon \text{DFD}(\mathbf{s}, \hat{\mathbf{d}}^i) \nabla_{\mathbf{s}} f_{t-\Delta t}(\mathbf{s} - \hat{\mathbf{d}}^i), \quad (16-2)$$

بعنوان یک روش تفاضلی، این تخمین به گرادیان فضایی بسیار وابسته است. به همین دلیل، گاهی اوقات روش های پیکسل بازگشتی بعنوان زیرمجموعه روش های گرادیانی یا تفاضلی در نظر گرفته می شوند.

روش تکرار شونده معادله (16-2) معمولاً بصورت پیکسل به پیکسل اعمال می شود، که یک ناحیه حرکت متراکم^۱، $\hat{\mathbf{d}}(\mathbf{s})$ ، را نتیجه می دهد. ممکن است تکرارها بصورت اسکن خطی، از یک خط به خط دیگر، یا از یک فریم به فریم دیگر انجام گیرند. به منظور کم کردن اثر نویز، عبارت به روز رسانی می تواند روی ناحیه $A = \{\mathbf{s}_1, \dots, \mathbf{s}_p\}$ بصورت زیر ارزیابی شود:

$$\hat{\mathbf{d}}^{i+1} = \hat{\mathbf{d}}^i - \varepsilon \sum_{j=1}^p W_j \text{DFD}(\mathbf{s}_j, \hat{\mathbf{d}}^i) \nabla_{\mathbf{s}} f_{t-\Delta t}(\mathbf{s}_j - \hat{\mathbf{d}}^i), \quad (17-2)$$

که $\sum_{j=1}^p W_j = 1$ و $W_j \geq 0$ می باشد.

همگرایی این روش وابستگی زیادی به اندازه گام ثابت ε دارد. مقدار زیاد ε به همگرایی سریع اما دقت کم، و در عوض مقدار کم آن به همگرایی کندتر اما دقت تخمین بیشتر منجر می شود. بنابراین، مطلوب است بین این دو مصالحه صورت گیرد. تعدادی الگوریتم به منظور بهبود عملکرد الگوریتم های پیکسل بازگشتی معرفی شده اند [۹]. اکثر آنها بر مبنای ایده جایگزینی اندازه گام ثابت ε با اندازه گام متغیر به منظور تطبیق بهتر با خواص آماری تصویر محلی و در نتیجه، همگرایی سریعتر و دقت بیشتر می باشند.

^۱Dense motion field

ناحیه حرکت متراکم روش های پیکسل بازگشتی می تواند بر مشکل دقت غلبه کند. با این حال، اطلاعات مازاد^۱ حرکتی زیادی را ایجاد می کند. برای غلبه بر این مشکل، عبارت به روز رسانی می تواند از یک تکرار به تکرار بعدی تنها بر اساس دیتای ارسال شده قبلی باشد. در این حالت، رمزگشا می تواند جابجایی های مشابه تولید شده در رمزگذار را تخمین بزند و نیازی به ارسال اطلاعات حرکت نمی باشد. با این حال، این روش قابلیت پیشگویی را کاهش می دهد. بعلاوه، پیچیدگی رمزگشا را افزایش می دهد.

نقص دیگر روش های پیکسل بازگشتی این است که آن ها به سادگی به کمینه محلی داخل سطح خطا همگرا می شوند. بعلاوه، نواحی با شدت روشنایی ملایم، ناپیوستگی ها درون حوزه حرکت، و جابجایی های بزرگ نمی توانند بطور مناسب بررسی شوند. [۱۰]

۲-۴ روش های حوزه فرکانس

روش های تخمین حرکت حوزه فرکانس^۲ بر مبنای این ویژگی تبدیل فوریه^۳ (FT) که جابجایی در حوزه فضایی با جابجایی خطی فاز در حوزه فرکانس متناظر است، عمل می کنند. بنابراین، فرض می کنیم که شدت های روشنایی تصویر فریم فعلی، f_t ، و فریم مرجع، $f_{t-\Delta t}$ ، روی ناحیه حرکت، A ، تنها به دلیل جابجایی، (d_x, d_y) ، تفاوت دارند. آنگاه

$$f_t(x, y) = f_{t-\Delta t}(x - d_x, y - d_y), \quad (x, y) \in A \quad (18-2)$$

با گرفتن تبدیل فوریه از هر دو طرف تساوی با توجه به متغیرهای فضایی (x, y) معادله حوزه فرکانس بر حسب متغیرهای فرکانسی (w_x, w_y) بدست می آید:

$$F_t(w_x, w_y) = F_{t-\Delta t}(w_x, w_y) e^{j(-w_x d_x - w_y d_y)} \quad (19-2)$$

^۱Overhead

^۲Frequency-domain method

^۳Fourier Transform

که F_t و $F_{t-\Delta t}$ به ترتیب تبدیل های فوریه فریم های فعلی و مرجع هستند.

اگر $\Delta\phi(w_x, w_y)$ را بعنوان اختلاف فاز بین تبدیل فوریه فریم فعلی و فریم مرجع تعریف کنیم، آنگاه

$$\begin{aligned} e^{j\Delta\phi(w_x, w_y)} &= e^{j[\phi_t(w_x, w_y) - \phi_{t-\Delta t}(w_x, w_y)]} \\ &= e^{j\phi_t(w_x, w_y)} \cdot e^{-j\phi_{t-\Delta t}(w_x, w_y)} \\ &= \frac{F_t(w_x, w_y)}{|F_t(w_x, w_y)|} \cdot \frac{F_{t-\Delta t}^*(w_x, w_y)}{|F_{t-\Delta t}^*(w_x, w_y)|} \quad (20-2) \end{aligned}$$

که ϕ_t و $\phi_{t-\Delta t}$ به ترتیب مؤلفه های فاز F_t و $F_{t-\Delta t}$ ، و بالانویس * مزدوج مختلط را نشان می دهد. اگر $c_{t,t-\Delta t}(x, y)$ را معکوس تبدیل فوریه $e^{j\Delta\phi(w_x, w_y)}$ تعریف کنیم، آنگاه

$$\begin{aligned} c_{t,t-\Delta t}(x, y) &= \mathcal{F}^{-1}\{e^{j\Delta\phi(w_x, w_y)}\} \\ &= \mathcal{F}^{-1}\{e^{j\phi_t(w_x, w_y)} \cdot e^{-j\phi_{t-\Delta t}(w_x, w_y)}\} \\ &= \mathcal{F}^{-1}\{e^{j\phi_t(w_x, w_y)}\} \otimes \mathcal{F}^{-1}\{e^{-j\phi_{t-\Delta t}(w_x, w_y)}\} \quad (21-2) \end{aligned}$$

که $\otimes$ عملگر کانولوشن دوبعدی است. به عبارت دیگر، $c_{t,t-\Delta t}(x, y)$ همبستگی متقابل معکوس تبدیل های فوریه مؤلفه های فاز F_t و $F_{t-\Delta t}$ می باشد. به همین دلیل، $c_{t,t-\Delta t}(x, y)$ تابع همبستگی فاز نامیده می شود. اهمیت این تابع زمانی مشخص می شود که آن را بر حسب اختلاف فاز در معادله (2-19) بنویسیم:

$$\begin{aligned} c_{t,t-\Delta t}(x, y) &= \mathcal{F}^{-1}\{e^{j\Delta\phi(w_x, w_y)}\} \\ &= \mathcal{F}^{-1}\{e^{j(-w_x d_x - w_y d_y)}\} \\ &= \delta(x - d_x, y - d_y) \quad (22-2) \end{aligned}$$

بنابراین، سطح همبستگی فاز یک ضربه مشخص در (d_x, d_y) دارد. این مشاهده، ایده پایه روش تخمین حرکت همبستگی فاز است. در این روش، معادله (2-20) برای محاسبه $c_{t,t-\Delta t}(x, y)$ استفاده می شود، سپس معکوس تبدیل فوریه برای بدست آوردن $c_{t,t-\Delta t}(x, y)$ اعمال می شود، و موقعیت آشکار شده ضربه در این تابع (d_x, d_y) را تخمین می زند.

روش تخمین حرکت همبستگی فاز ابتدا توسط کوگلین^۱ و هاینز^۲ ر ۱۹۷۵ ارائه شد [۱۱]. سپس بطور گسترده توسط توماس^۳ مورد مطالعه قرار گرفت [۱۲]. در این مطالعه، توماس ویژگی‌های تابع همبستگی فاز را تحلیل کرد. او استفاده از تابع وزن دار را برای هموار کردن سطح همبستگی و از بین بردن قله‌های نادرست پیشنهاد داد.

روش همبستگی فاز پیچیدگی محاسباتی کمی به ویژه با استفاده از تبدیل‌های فوریه سریع^۴ (FFTs) دارد. بعلاوه، نسبتاً به تغییرات شدت روشنایی حساس است، زیرا جابجایی در مقدار میانگین یا ضرب در یک مقدار ثابت اثری روی فاز فوریه ندارد. همچنین، این روش می‌تواند چندین شیء متحرک را شناسایی کند، زیرا آن‌ها بصورت چندین قله در سطح همبستگی ظاهر می‌شوند. علاوه بر استفاده در رمزگذاری ویدئو، روش همبستگی فاز بطور موفقیت آمیز با تجهیزات تبدیل استانداردهای تجاری ترکیب شده است [۱۳].

۲-۵ روش‌های تطبیق بلوک

تخمین حرکت تطبیق بلوک^۵ (BMME) بطور گسترده در روش تخمین حرکت برای رمزگذاری ویدئو مورد استفاده قرار گرفته است. ابتدا این روش توسط جی. آر. جین^۶ و آ. کا. جین^۷ در ۱۹۸۱ معرفی شد [۱۴]. در الگوریتم تطبیق بلوک^۸ (BMA) آن‌ها، ابتدا فریم فعلی، f_t ، به بلوک‌های $M \times N$ پیکسل تقسیم می‌شود. سپس این الگوریتم فرض می‌کند که همه پیکسل‌های داخل بلوک در یک جهت مشابه حرکت می‌کنند. بنابراین، بردار حرکت مشابه‌ای، $\mathbf{d}=[d_x, d_y]^T$ ، به همه پیکسل‌های داخل بلوک اختصاص داده می‌شود. این بردار

^۱ Kuglin

^۲ Hines

^۳ Thomas

^۴ Fast Fourier Transforms

^۵ Block-Matching Motion Estimation

^۶ J. R. Jain

^۷ A. K. Jain

^۸ Block-Matching Algorithm

حرکت بصورت جستجوی بهترین بلوک منطبق در پنجره جستجوی^۱ بزرگتر $(M + 2d_{m_x}) \times (N + 2d_{m_y})$ پیکسل متمرکز شده در موقعیت مشابه در فریم مرجع، $f_{t-\Delta t}$ تخمین زده می شود که در آن d_{m_x} و d_{m_y} به ترتیب حداکثر جابجایی حرکتی مجاز در جهت های افقی و عمودی می باشند. این فرآیند در شکل ۱-۲ نشان داده شده است و بصورت زیر فرمول بندی شده است:

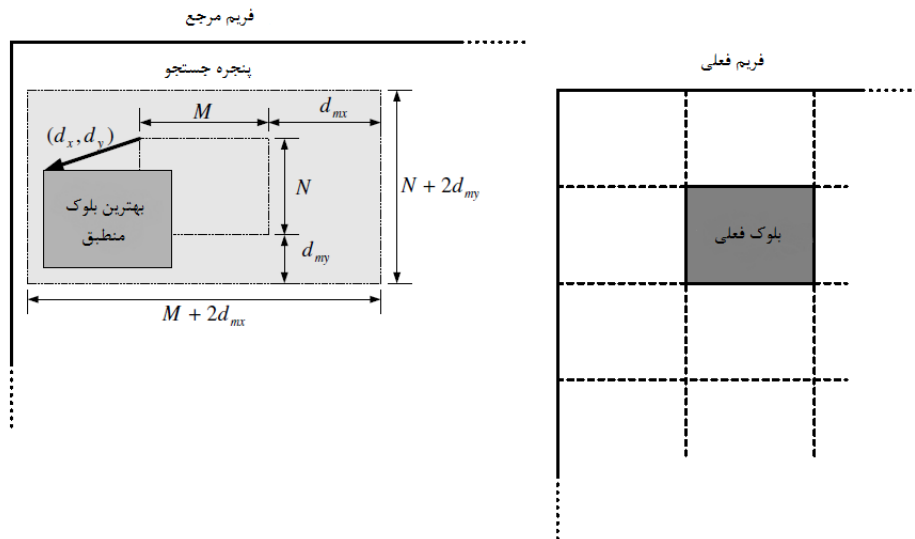
$$(\hat{d}_x, \hat{d}_y) = \underset{i,j}{\operatorname{argmin}} \text{BDM}(i,j), \text{ که } |i| \leq d_{m_x} \text{ و } |j| \leq d_{m_y} \quad (2-23)$$

و $\text{BDM}(i,j)$ معیار انحراف بلوک^۲ است که کیفیت تطبیق را بین بلوک در فریم فعلی و بلوک منتخب در فریم مرجع که با جابجایی (i,j) انتقال یافته است، اندازه می گیرد. استفاده از بلوک های مربعی $N \times N$ پیکسلی و حداکثر جابجایی حرکتی $\pm d_m$ در هر دو جهت بسیار رایج است. هنگامیکه معادله (۲-۲۳) برای همه جابجایی های (i,j) ممکن (یعنی، برای همه بلوک های منتخب در پنجره جستجو) ارزیابی شود، BMA بعنوان الگوریتم جستجوی کامل^۳ (FS) شناخته می شود.

^۱Search window

^۲Block Distortion Measure

^۳Full Search



شکل ۲-۱: تخمین حرکت تطبیق بلوک [۱]

در زیربخش های زیر، پارامترهای مختلف BMA معرفی شده و اهمیت آن ها روی عملکرد ارزیابی می شود.

۲-۵-۱ تابع تطبیق

تابع تطبیق^۱ (یا BDM) می تواند هر تابعی که انحراف یا تطبیق بین بلوک در فریم فعلی و بلوک منتخب جابجا شده در فریم مرجع را اندازه می گیرد، باشد. انتخاب BDM مناسب بسیار مهم است، زیرا آن در کیفیت پیشگویی و پیچیدگی محاسباتی الگوریتم نقش مهمی دارد.

یک تابع تطبیق، تابع همبستگی متقاطع نرمالیزه شده^۲ (NCCF) است و بصورت زیر تعریف می شود:

$$NCCF(i, j) = \frac{\sum_{(x,y) \in \mathbb{R}} f_t(x,y) \cdot f_{t-\Delta t}(x-i, y-j)}{\sqrt{\sum_{(x,y) \in \mathbb{R}} f_t^2(x,y)} \cdot \sqrt{\sum_{(x,y) \in \mathbb{R}} f_{t-\Delta t}^2(x-i, y-j)}} \quad (2-24)$$

^۱ Matching function

^۲ Normalized Cross-Correlation Function

از آنجایی که فرآیند تخمین حرکت به حداقل کردن سیگنال DFD کمک می‌کند، یک انتخاب طبیعی برای تابع تطبیق متوسط مجذور خطا^۱ می‌باشد، که اغلب بصورت مجموع مجذور اختلافات^۲ (SSD) فرمول بندی می‌شود:

$$SSD(i, j) = \sum_{(x, y) \in \mathcal{R}} (f_t(x, y) - f_{t-\Delta t}(x - i, y - j))^2 \quad (25-2)$$

تابع تطبیق بسیار مشابه دیگر، مجموع قدرمطلق اختلافات^۳ (SAD) می‌باشد:

$$SAD(i, j) = \sum_{(x, y) \in \mathcal{R}} |f_t(x, y) - f_{t-\Delta t}(x - i, y - j)| \quad (26-2)$$

به منظور مقایسه عملکرد این توابع تطبیق، BMA با جستجوی تمام پیکسلی اعمال می‌شود. الگوریتم از بلوک‌های 16×16 و حداکثر جابجایی حرکتی ± 15 در هر دو جهت استفاده می‌کند. در این الگوریتم، حرکت با استفاده از فریم‌های قبلی تخمین و جبران می‌شود، و بردارهای حرکت به گونه‌ای محدود می‌شوند که به خارج از فریم مرجع اشاره نکنند. شکل ۲-۲ عملکرد الگوریتم با توابع تطبیق مختلف را هنگامیکه برای ۱۰ فریم اول رشته سرکارگر^۴ با آهنگ فریم^۵ ۸.۳۳ فریم بر ثانیه اعمال می‌شود، نشان می‌دهد. مقادیر نسبت اوج سیگنال به نویز^۶ (PSNR) تنها برای مؤلفه روشنایی می‌باشند. از این شکل متوجه می‌شویم که معیار SSD عملکرد بهتری دارد، و عملکرد SAD بسیار به آن نزدیک است. از سوی دیگر، معیار NCCF بدترین عملکرد را دارد [۱]. جدول ۲-۱ عملکرد این الگوریتم را بر حسب پیچیدگی محاسباتی مقایسه می‌کند. با توجه به جدول درمی‌یابیم که معیار SAD کمترین پیچیدگی محاسباتی را دارد، زیرا هیچ ضربی را شامل نمی‌شود. به دلیل کیفیت پیشگویی خوب و پیچیدگی محاسباتی کم، SAD در بیشتر پیاده‌سازی‌ها استفاده می‌شود. در همه نتایج بعدی SAD بعنوان تابع تطبیق در نظر گرفته می‌شود.

^۱Mean Squared Error

^۲Sum of Squared Differences

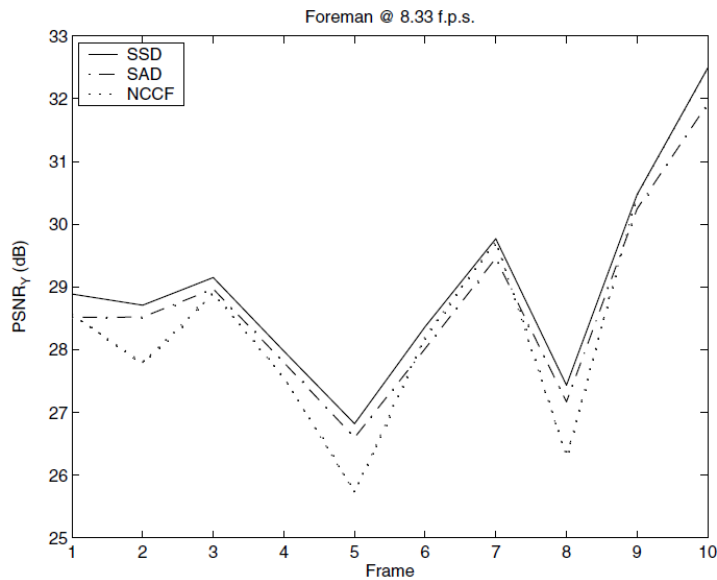
^۳Sum of Absolute Differences

^۴Foreman

^۵آهنگ فریم بصورت تعداد فریم‌های اسکن شده در یک ثانیه تعریف می‌شود.

^۶Peak Signal to Noise Ratio

توابع تطبیق بسیاری وجود دارد. اکثر آن‌ها تلاش می‌کنند پیچیدگی را کاهش دهند، اما اغلب کیفیت پیشگویی نیز کاهش می‌یابد.



شکل ۲-۲: کیفیت بازسازی برای SSD، SAD و NCCF [۱]

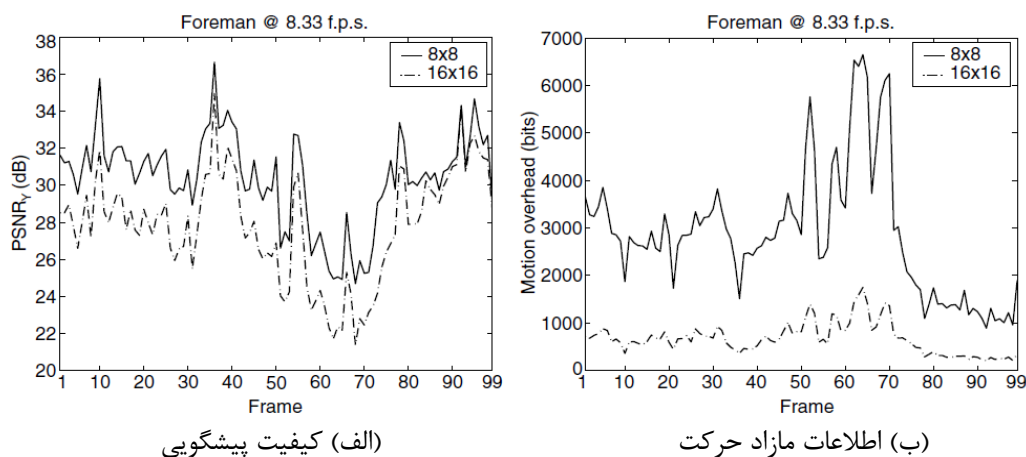
جدول ۲-۱: پیچیدگی محاسباتی SSD، SAD و NCCF برای بلوک $N \times N$

	SAD	SSD	NCCF
$ \cdot $	N^2	—	—
—	N^2	N^2	—
+	$N^2 - 1$	$N^2 - 1$	$3(N^2 - 1)$
$\times$	—	N^2	$3N^2 + 1$
$\div$	—	—	1
$\sqrt{\quad}$	—	—	2

۲-۵-۲ اندازه بلوک

پارامتر مهم دیگر BMA اندازه بلوک است. شکل ۳-۲ عملکرد BMA را برای اندازه‌های مختلف 8×8 و 16×16 نشان می‌دهد. از شکل ۳-۲ (الف) می‌توان دید که اندازه بلوک

کوچکتر کیفیت پیشگویی بهتری را نتیجه می‌دهد [۱]. برای این امر چند دلیل وجود دارد. اندازه بلوک کوچکتر مشکل دقت را کمتر می‌کند. به عبارت دیگر، با اندازه بلوک کوچکتر، احتمال کمتری وجود دارد که بلوک، حاوی شیء‌های متحرک در جهات مختلف باشد. از آنجایی که اندازه بلوک کوچکتر به معنی وجود بلوک‌های بیشتر (و در نتیجه بردارهای حرکت بیشتر) در یک فریم می‌باشد، کیفیت پیشگویی با هزینه اطلاعات مزاد حرکت بیشتر بدست می‌آید، که در شکل ۲-۳ (ب) می‌توان دید. اکثر استانداردهای رمزگذاری ویدئو از اندازه بلوک 16×16 به منظور برقراری مصالحه بین کیفیت پیشگویی و اطلاعات مزاد حرکت، استفاده می‌کنند.

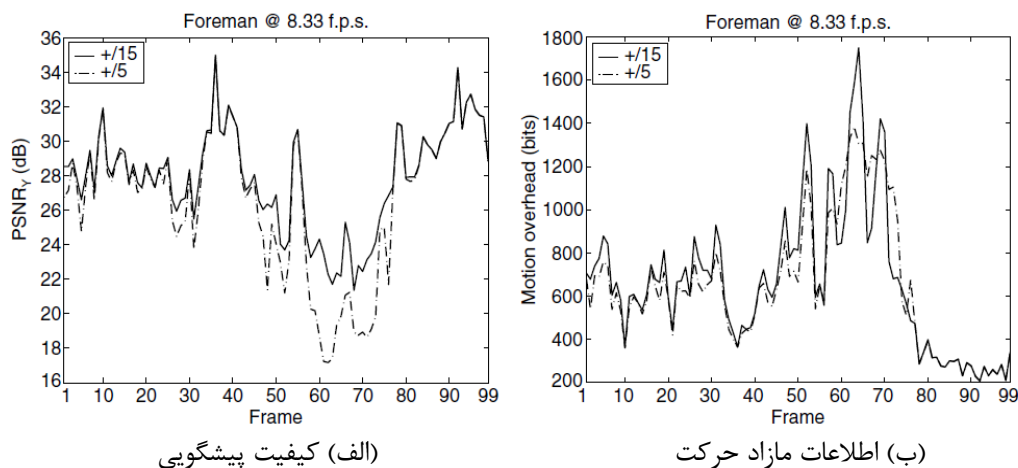


شکل ۲-۳: عملکرد BMA با اندازه‌های بلوک مختلف [۱]

۲-۵-۳ محدوده جستجو

حداکثر جابجایی حرکت مجاز d_m ، که محدوده جستجو نام دارد، تأثیر مستقیمی روی پیچیدگی محاسبات و کیفیت پیشگویی BMA دارد. d_m کوچک برای نواحی با حرکت سریع جبران سازی ضعیف و در نتیجه کیفیت پیشگویی ضعیفی دارد. این امر از شکل ۲-۴ (الف)، که عملکرد دو محدوده ± 5 و ± 15 را مقایسه می‌کند، بدیهی به نظر می‌رسد. از سوی دیگر،

d_m بزرگ کیفیت پیشگویی بهتری را نتیجه می‌دهد، اما پیچیدگی محاسباتی را نیز افزایش می‌دهد (زیرا $(2d_m+1)^2$ بلوک در پنجره جستجو برای تطبیق وجود دارد). هم‌چنین d_m بزرگتر، بردارهای حرکت بزرگتر و افزایش کم اطلاعات مازاد را نتیجه می‌دهد، که در شکل ۴-۲ (ب) مشاهده می‌شود.



شکل ۴-۲: عملکرد BMA با محدوده جستجوی مختلف [۱]

بطور کلی، حداکثر جابجایی مجاز $d_m = \pm 15$ پیکسل برای کاربردهای با آهنگ بیت کم مناسب است.

۴-۵-۲ دقت جستجو

ابتدا، BMA برای تخمین جابجایی‌های حرکت با دقت تمام پیکسل طراحی شده بود. واضح است که این طراحی عملکرد الگوریتم را محدود می‌کند، زیرا در حقیقت حرکت اشیاء کاملاً به شبکه نمونه برداری ارتباطی ندارد. به منظور بسط BMA با دقت زیرپیکسل^۱ پیشنهاداتی ارائه شده است. برای مثال، اریکسون^۲ [۱۵] ثابت کرد که بهره پیشگویی حدود

^۱Sub-pixel
^۲Ericsson

۲dB را می توان با دقت ۱/۸ پیکسل بدست آورد. گیرود^۱ [۱۶] یک تحلیل نظری پیشگویی حرکت جبران شده با دقت زیرپیکسل را نشان داد. او نتیجه گرفت که با اندازه بلوک 16×16 ، برای سیگنال های تلویزیون دقت ربع پیکسل^۲ و برای سیگنال های تلفن ویدئویی دقت نصف پیکسل^۳ مناسب است. امروزه، اکثر استانداردهای رمزگذاری ویدئو دقت نصف پیکسل را پذیرفته اند [۱].

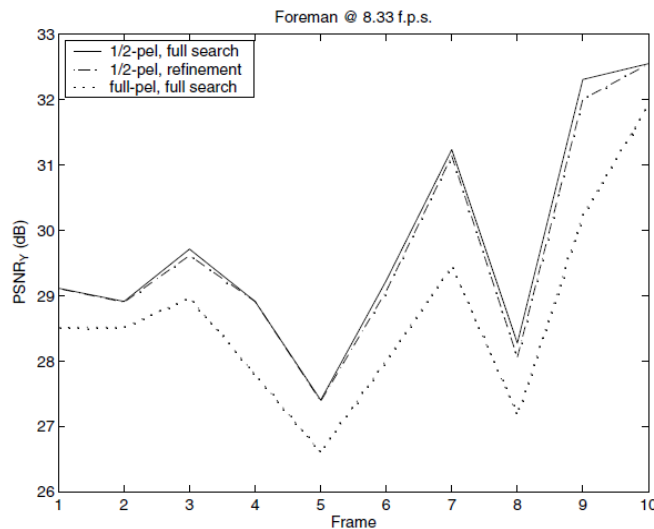
با این حال، باید خاطر نشان کرد که بهبود کیفیت پیشگویی با دقت زیرپیکسل، پیچیدگی محاسباتی را بطور قابل توجهی افزایش می دهد. این افزایش به دو دلیل است. اول، شدت های روشنایی فریم مرجع باید در موقعیت های زیرپیکسل قرار بگیرد. دوم، در این حالت در پنجره جستجو بلوک های منتخب زیادی وجود خواهد داشت. بعنوان مثال، هنگامیکه دقت از تمام پیکسل به نصف پیکسل تغییر می کند، تعداد بلوک های منتخب در پنجره جستجو از $(2d_m+1)^2$ به $(4d_m+1)^2$ افزایش می یابد. برای کم کردن این پیچیدگی، اکثر رمزگذار/رمزگشاهای ویدئو دقت زیرپیکسل را بعنوان مرحله پس پردازش^۴ انجام می دهند، که در آن ابتدا بردار حرکت تمام پیکسل با استفاده از جستجوی کامل بدست می آید و سپس این بردار با استفاده از جستجوی محدود شده برای دقت زیرپیکسل اصلاح می شود.

^۱ Girod

^۲ Quarter-pixel

^۳ Half-pixel

^۴ Postprocessing



شکل ۲-۵: عملکرد BMA با دقت زیرپیکسل [۱]

این امر پیچیدگی محاسباتی را بسیار کم می‌کند در حالیکه کیفیت پیشگویی را تغییر نمی‌دهد، که در شکل ۲-۵ نشان داده شده است.

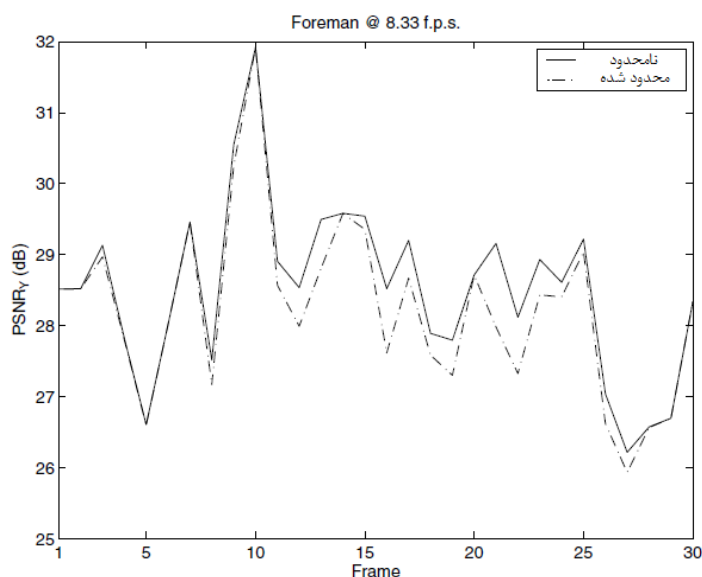
۲-۵-۵ بردارهای حرکت نامحدود

در برخی موارد (برای مثال در بلوک‌های مرزی) بخشی از پنجره جستجو خارج از محدوده فریم مرجع قرار می‌گیرد. به این معنی که تعدادی از بلوک‌های منتخب در پنجره جستجو بصورت جزئی یا کامل خارج از فریم مرجع قرار می‌گیرند. دو راه برای بررسی چنین بلوک‌هایی وجود دارد. در روش بردارهای حرکت محدود شده^۱، از این بلوک‌ها در طی تخمین حرکت صرف‌نظر می‌شود. با این حال، در روش بردارهای حرکت نامحدود^۲ این بلوک‌ها در فرآیند تخمین و جبران حرکت شامل می‌شوند. در این حالت، پیکسل مورد نظر خارج از فریم معمولاً توسط نزدیکترین پیکسل مرزی تقریب زده می‌شود. این روش نامحدود

^۱Restricted motion vectors

^۲Unrestricted motion vectors

می‌تواند کیفیت پیشگویی را در مرزهای فریم، به ویژه در هنگام حرکت دوربین یا پس‌زمینه بهبود بخشد. این روش مخصوصاً برای فرمت‌های فریم کوچک، که در آن بلوک‌های مرزی درصد بالایی از محدوده فریم را نشان می‌دهد، مناسب است. شکل ۲-۶ این بهبود را برای بخشی از رشته سرکارگر نشان می‌دهد.



شکل ۲-۶: عملکرد BMA با بردارهای حرکت محدود شده و نامحدود [۱]

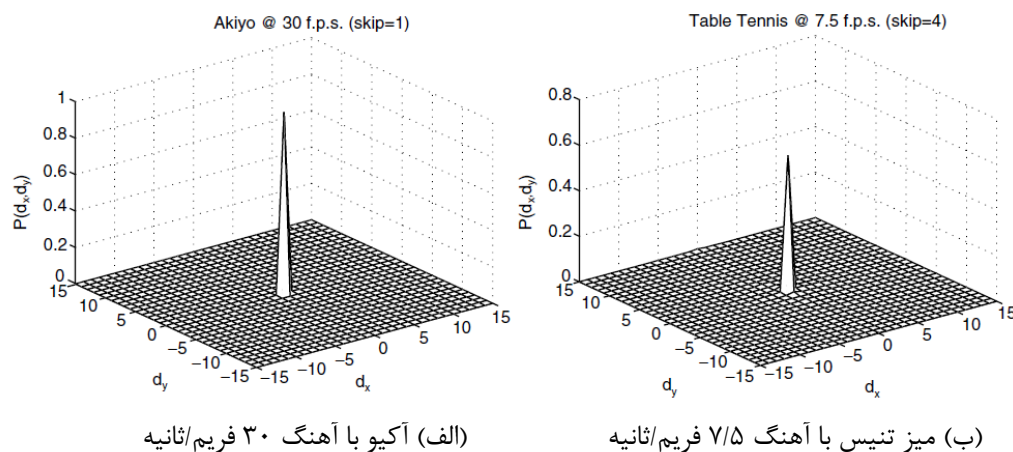
۲-۵-۶ ویژگی‌های ناحیه‌ها و سطوح خطای تطبیق بلوک

این زیربخش تعدادی از ویژگی‌های اصلی الگوریتم BMME بکار برده شده برای رشته‌های ویدئویی را بیان می‌کند. تمام اشکال در این زیربخش با استفاده از الگوریتم تطبیق بلوک جستجوی کامل با بلوک‌های 16×16 ، حداکثر جابجایی ± 15 پیکسل، بردارهای حرکت محدود شده، معیار انحراف بلوک SAD، و فریم‌های مرجع اصلی تولید شده‌اند.

ویژگی ۱. توزیع ناحیه حرکتی بلوک^۱ به مرکز متمایل است. به این معنا که جابجایی‌های کوچکتر محتمل‌تر هستند و بردار حرکت (۰،۰) بیشترین احتمال وقوع را دارد.

^۱Block-motion field

به عبارت دیگر، اکثر بلوک‌ها ثابت یا نیمه ثابت هستند. این ویژگی در شکل ۷-۲ (الف) برای آکیو^۱ با آهنگ ۳۰ فریم/ثانیه نشان داده شده است. این ویژگی برای رشته‌های با محتوای حرکت بیشتر و نرخ فریم کمتر نیز صادق است، که در شکل ۷-۲ (ب) برای میز تنیس^۲ با آهنگ ۷/۵ فریم/ثانیه نشان داده شده است.



شکل ۷-۲: توزیع متمرکز شده در مرکز ناحیه حرکتی بلوک [۱]

ویژگی ۲. ناحیه حرکتی بلوک هموار است و به آرامی تغییر می‌کند. به عبارت دیگر، همبستگی زیادی بین بردارهای حرکت بلوک‌های همسایه وجود دارد. بنابراین، پیدا کردن بلوک‌های همسایه با بردارهای حرکت تقریباً مشابه بسیار رایج است.

ویژگی ۳. معمولاً سطح خطا چندوجهی^۳ است. در بیشتر موارد، سطح خطا شامل یک یا چند کمینه محلی است، که در شکل ۸-۲ نشان داده شده است.

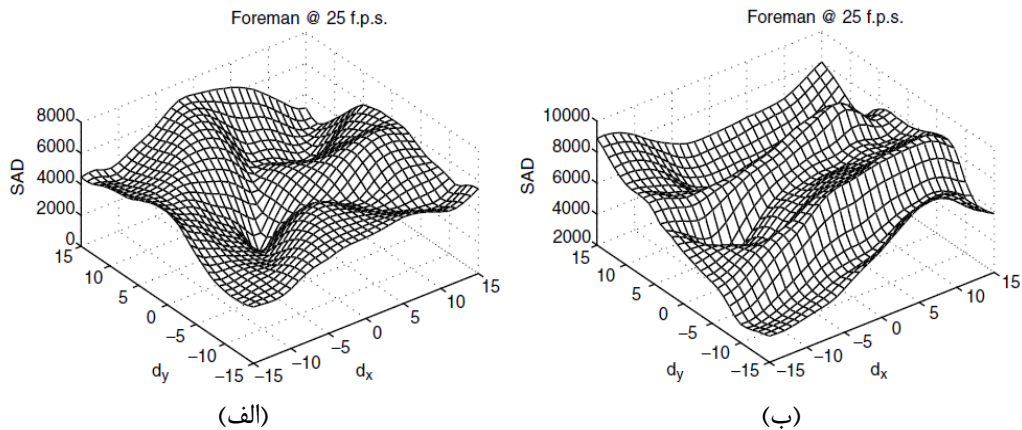
ویژگی ۴. مقدار کمینه اصلی یک سطح خطا بر طبق دلایلی چون محتوای حرکت، محتوای بلوک می‌تواند تغییر کند. بعنوان مثال، شکل ۸-۲ سطح خطای دو بلوک از یک

^۱ Akiyo

^۲ Table Tennis

^۳ Multimodal

فریم مشابه را نشان می‌دهد. مقدار کمینه اصلی سطح در شکل ۸-۲ (الف) ۶۱۴ و در شکل ۸-۲ (ب) ۳۱۵۴ می‌باشد.



شکل ۸-۲: نمونه‌هایی از سطوح خطای چندوجهی [۱]

۲-۶ مقایسه الگوریتم‌ها

در این بخش نتایج مقایسه روش‌های تخمین حرکت بحث شده در بخش‌های ۲-۲ تا ۲-۵ ارائه می‌شود. هدف اصلی این مقایسه پاسخ به این سوال است: بهترین الگوریتم تخمین حرکت برای رمزگذاری ویدئو کدام است؟ در این مقایسه، الگوریتم‌های زیر بررسی شده‌اند [۱]:

DFA^۱ پیاده‌سازی روش تفاضلی کافوریو و رکا در معادله (۲-۹) است. در این الگوریتم، ناحیه حرکت، A، برای بلوک‌های ۱۶×۱۶ پیکسل بدست می‌آید.

^۱Differential Frame Algorithm

PRA^۱ پیاده سازی الگوریتم پیکسل بازگشتی نتروالی و رابینز در معادله (۲-۱۷) است. در این الگوریتم، بردار حرکت پیکسل در خط قبلی بعنوان تخمین حرکت اولیه، $\hat{d}^i$ ، پیکسل فعلی در نظر گرفته می شود، اندازه گام به روز رسانی $\varepsilon = 1/1024$ انتخاب شده است، عبارت به روز رسانی روی ناحیه 3×3 پیکسلی متمرکز شده حول پیکسل فعلی محاسبه و میانگین گیری شده، و پنج تکرار برای هر پیکسل انجام شده است.

PCA^۲ پیاده سازی روش همبستگی فاز در معادله (۲-۲۰) و (۲-۲۱) است. در این الگوریتم، پنجره 32×32 پیکسلی متمرکز شده حول بلوک 16×16 فعلی به منظور ایجاد سطح همبستگی فاز مورد استفاده قرار گرفته است. اکثراً سه نقطه اوج بارز در این سطح آشکار شده و جابجایی های کوچک در نظر گرفته نمی شوند. سپس سه جابجایی مورد نظر با استفاده از SAD بین بلوک فعلی و بلوک مورد نظر در فریم مرجع بررسی می شوند. جابجایی با کمترین SAD بعنوان بردار حرکت بلوک فعلی انتخاب می شود.

BMA^۳ پیاده سازی الگوریتم تطبیق بلوک با جستجوی کامل است. در این الگوریتم، اندازه بلوک 16×16 و معیار تطبیق SAD می باشد.

در همه الگوریتم ها، حداکثر جابجایی مجاز ± 15 پیکسل در هر جهت است و بردارهای حرکت مجازند که به خارج از فریم مرجع نیز اشاره کنند (یعنی بردارهای حرکت نامحدود). پارامترهای شبیه سازی مختلف نتایج متفاوتی را نتیجه می دهد. برای مثال، عملکرد PRA با افزایش تعداد تکرارها بهبود می یابد اما پیچیدگی محاسبات را افزایش می دهد. این امر برای PCA هنگامیکه نقاط اوج بیشتری را امتحان می کند، نیز صادق است.

^۱Pixel-Recursive Algorithm

^۲Phase Correlation Algorithm

^۳Block Matching Algorithm

شکل ۲-۹ کیفیت پیشگویی چهار الگوریتم را هنگامیکه روی سه رشته ویدئویی آکیو، سرکارگر، و میز تنیس با آهنگ فریم متفاوت اعمال می‌شود، مقایسه می‌کند.

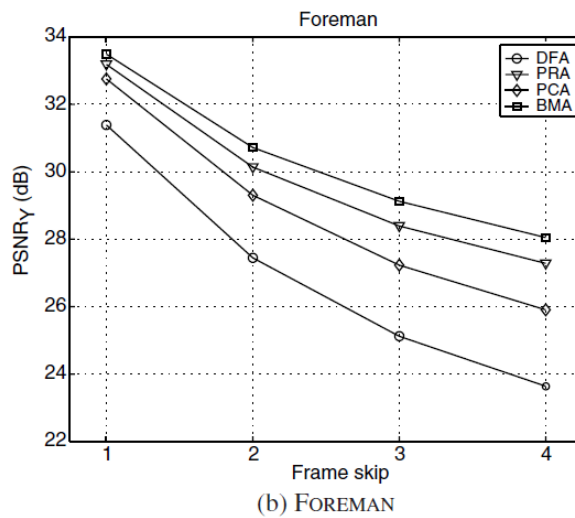
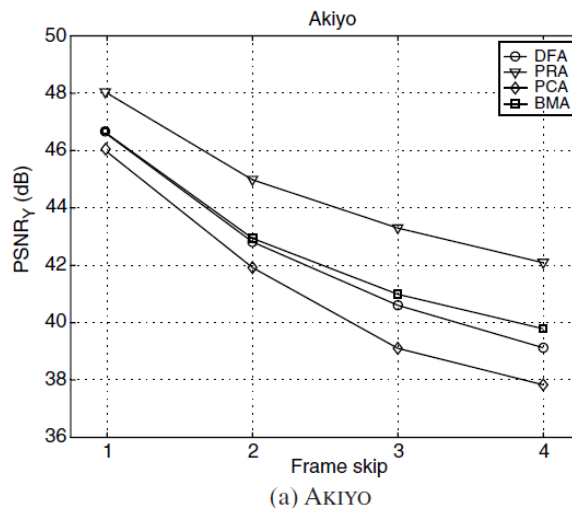
همانطور که انتظار می‌رود، DFA برای رشته‌های با جابجایی کم (آکیو) و آهنگ فریم زیاد بخوبی عمل می‌کند. برای رشته‌های با جابجایی زیاد (سرکارگر و میز تنیس) و آهنگ فریم کم بردار حرکت بزرگتر شده، کیفیت تقریب سری‌های تیلور ضعیف، و عملکرد DFA بد می‌شود.

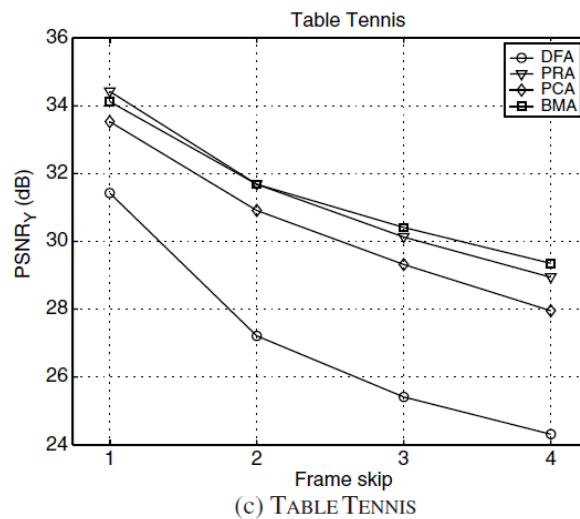
به دلیل ناحیه حرکت متراکم PRA، این روش عملکرد بهتری برای آکیو و عملکرد رقابتی برای سرکارگر و میز تنیس دارد. افت نسبی عملکرد برای رشته‌های با حرکت زیاد و آهنگ فریم کم به چند دلیل است. با بردارهای حرکت بزرگتر، احتمال بیشتری وجود دارد که الگوریتم قبل از اینکه به کمینه اصلی برسد، در کمینه محلی به دام بیفتد. همچنین، ممکن است حداکثر تعداد تکرارها برای رسیدن به کمینه اصلی کافی نباشد. با این حال، افزایش تعداد تکرارها پیچیدگی الگوریتم را بیشتر می‌کند.

بطور کلی، عملکرد PCA بین DFA و PRA است. عملکرد ضعیف برای آکیو ممکن است به دلیل نقاط اوج نادرست ایجاد شده توسط اثرات نشت طیفی و مرزی باشد. چنین اثراتی می‌تواند با بکار بردن یک تابع وزن‌دار به منظور هموار کردن سطح همبستگی فاز کاهش یابد.

بهترین عملکرد توسط BMA ارائه شده است. این روش صرفنظر از نوع رشته و آهنگ فریم بخوبی عمل می‌کند. در حقیقت، برای رشته‌های با جابجایی زیاد (سرکارگر و میز تنیس)، BMA عملکردی عالی از خود نشان داده است.

به دو دلیل مقایسه دو الگوریتم PRA و BMA جالب می‌باشد. اول، این دو الگوریتم بهترین کیفیت پیشگویی را بدست می‌دهند. دوم، آن‌ها دو روش مختلف را برای تخمین حرکت ارائه می‌دهند (پیکسل بازگشتی و مبتنی بر بلوک).





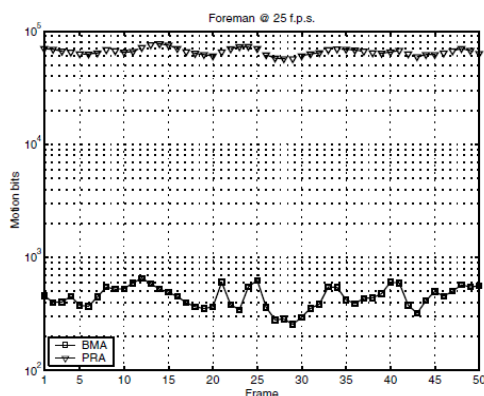
شکل ۲-۹: کیفیت پیشگویی الگوریتم‌های تخمین حرکت متفاوت [۱]

شکل ۲-۱۰ عملکرد PRA و BMA را برای ۵۰ فریم اول رشته ویدئوی سرکارگر با ۲۵ فریم/ثانیه مقایسه می‌کند. دو نسخه PRA در نظر گرفته می‌شود: PRA که همان الگوریتم توضیح داده شده است، و PRA-C که الگوریتم با عبارت به روز رسانی مبتنی بر بخش علی^۱ ناحیه ۵×۵ پیکسلی متمرکز شده حول پیکسل فعلی است. از آنجایی که PRA-C مبتنی بر دیتای علی است، هیچ اطلاعات مازاد حرکت برای این روش ارسال نمی‌شود. به دلیل میزان زیاد حرکت در سرکارگر، حداکثر تعداد تکرارها برای هر دو الگوریتم پیکسل بازگشتی ۱۰ می‌باشد.

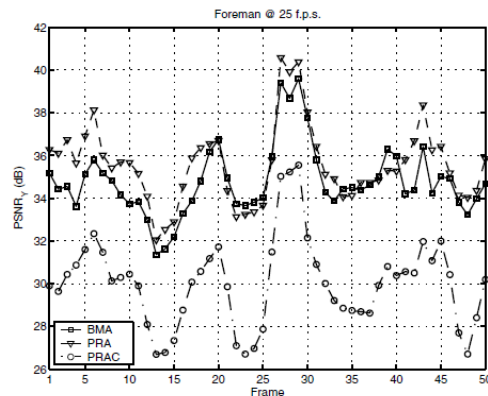
هدف تخمین حرکت در رمزگذاری ویدئو به حداقل رساندن آهنگ بیت پارامترهای حرکت (بیت‌های حرکت) و سیگنال خطای پیشگویی (بیت‌های DFD) می‌باشد. همانطور که در شکل ۲-۱۰ نشان داده شده است، سه الگوریتم، سه مصالحه مختلف را بین کیفیت پیشگویی و اطلاعات مازاد حرکت ارائه می‌دهند. به دلیل ناحیه متراکم حرکت در PRA، این الگوریتم بهترین کیفیت پیشگویی و در نتیجه حداقل بیت‌های DFD را دارا می‌باشد. با این

^۱Causal part

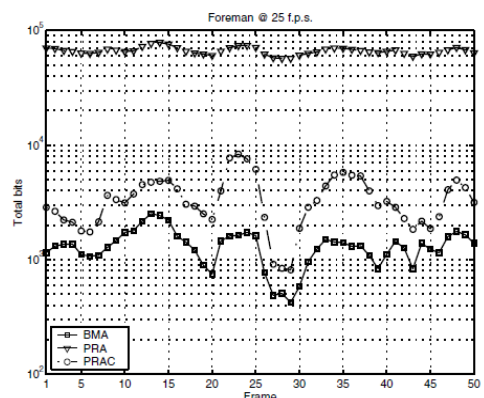
حال، اطلاعات مازاد حرکت زیادی دارد، که موجب افزایش بسیار زیاد آهنگ بیت نهایی می‌شود. پیاده‌سازی علی PRA، PRA-C، این روش را محدود کرده و بطور قابل توجهی کیفیت پیشگویی آن را کاهش می‌دهد. بنابراین، PRA-C اطلاعات مازاد حرکت را با هزینه افزایش بیت‌های DFD حذف می‌کند. بعلاوه، این پیاده‌سازی علی پیچیدگی رمزگشا را افزایش می‌دهد. بهترین مصالحه مربوط به BMA است. این الگوریتم از روش مبتنی بر بلوک به منظور کاهش اطلاعات مازاد حرکت استفاده می‌کند، در حالیکه کیفیت پیشگویی بسیار خوبی را نتیجه می‌دهد. از اینرو، این الگوریتم بطور گسترده در استانداردهای رمزگذاری ویدئو مورد استفاده قرار می‌گیرد.



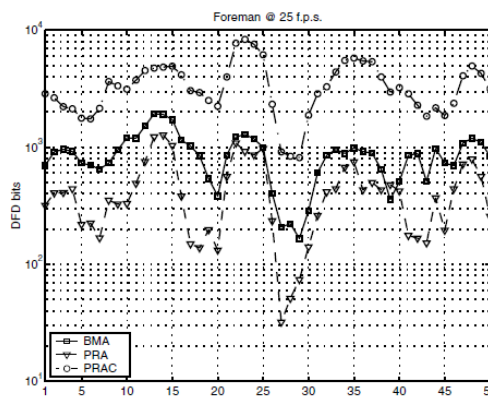
(ب) بیت‌های حرکت



(الف) کیفیت بازسازی



(د) بیت‌های نهایی=حرکت+DFD



(ج) بیت‌های DFD

شکل ۲-۱۰: مقایسه بین الگوریتم‌های تخمین حرکت BMA و PRA [۱]

کیفیت پیشگویی BMA در آهنگ بیت بسیار کم به ویژه برای رشته کم تحرک آکیو افت می کند و به الگوریتم های دیگر نزدیک می شود. با این حال، باز هم استفاده از BMA قابل توجیه است [۱].

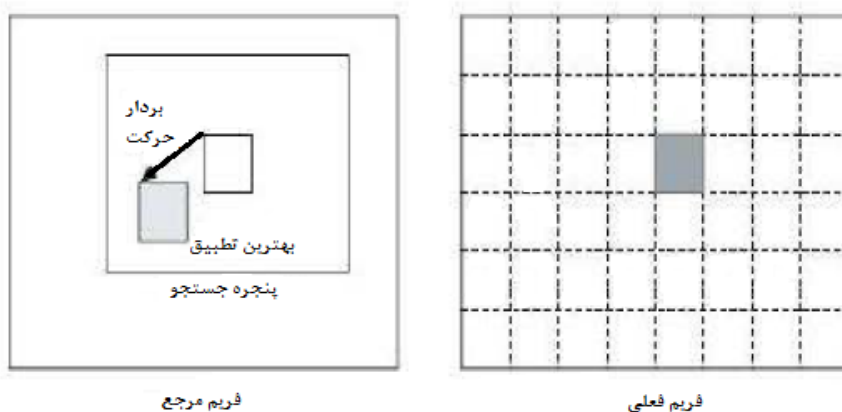
فصل سوم

مروری بر روش های تطبیق بلوک

۳ فصل سوم

۳-۱ مقدمه

الگوریتم تطبیق بلوک در شکل ۳-۱ نشان داده شده است. در الگوریتم تطبیق بلوک، هر فریم به چندین بلوک تقسیم می شود، که هر بلوک شامل بلوک های روشنایی^۱ و رنگ تابی^۲ است. معمولاً، تخمین حرکت تنها روی بلوک روشنایی عمل می کند. هر بلوک روشنایی در فریم فعلی با تعدادی بلوک در ناحیه جستجو روی فریم مرجع مطابقت دارد. این بلوک ها، نسخه های جابجا شده بلوک اصلی هستند. بهترین بلوک پیدا شده و جابجایی آن (بردار حرکت) ذخیره می شود. در رمزگذار، فریم ورودی از پیشگویی فریم مرجع تفریق می شود. سپس بردار حرکت و خطای بدست آمده به جای بلوک روشنایی اصلی ارسال می شود؛ بنابراین حشویات بین فریمی حذف شده و فشرده سازی دیتا بدست می آید. در گیرنده، رمزگشا سیگنال خطای فریم را از دیتای دریافت شده ساخته و به فریم های مرجع بازسازی شده اضافه می کند.



شکل ۳-۱: تخمین حرکت تطبیق بلوک [۱۹]

^۱Luminance
^۲Chrominance

در دو دهه اخیر روش های مختلفی برای BMA پیشنهاد شده است. همه این روش ها برای رسیدن به یک یا بیشتر از سه هدف کاهش پیچیدگی محاسبات، پیدا کردن حرکت درست (کیفیت بالاتر) و کاهش آهنگ بیت (آهنگ فشرده سازی بالاتر) پیشنهاد شده اند [۱۷]. در ادامه این فصل تعدادی از روش های تطبیق بلوک شرح داده می شود.

۲-۳ الگوریتم جستجوی کامل (FS)

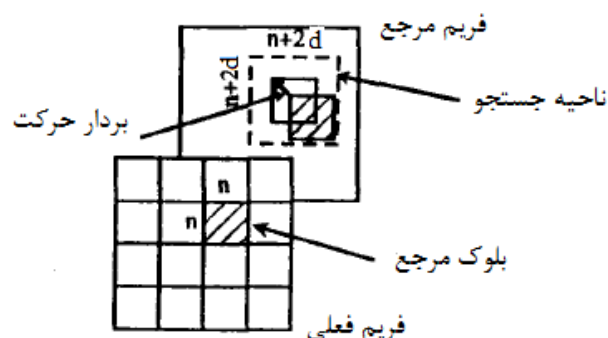
به منظور پیدا کردن بهترین بلوک منطبق در فریم مرجع، مقایسه بلوک فعلی با همه بلوک های فریم مرجع ضروری است. تخمین حرکت جستجوی کامل، مقدار مجموع قدرمطلق اختلاف (SAD) را در هر موقعیت ممکن در پنجره جستجو محاسبه می کند؛ عبارت دیگر الگوریتم جستجوی کامل^۱ (FS) همه بلوک ها را در پنجره جستجوی بزرگتر بررسی می نماید [۱۸ و ۱۹].

بلوک $N \times N$ پیکسل از فریم مورد بررسی و پنجره جستجو در محدوده $(N+2d) \times (N+2d)$ در فریم مرجع، که d حداکثر جابجایی مجاز است، در نظر گرفته شده، و در شکل ۲-۳ نشان داده شده است. برای هر یک از $(2d+1)^2$ موقعیت جستجو، بلوک انتخاب شده از پنجره جستجو با بلوک با اندازه $N \times N$ پیکسل مقایسه می شود. بهترین بلوک منطبق همراه با بردار حرکت بعد از بررسی همه $(2d+1)^2$ موقعیت جستجو بدست می آید. با اندازه بلوک 16×16 و حداکثر جابجایی ± 7 ، تعداد نقاط بررسی شده ۲۵۵ می باشد.

با بررسی کامل همه بلوک های داخل پنجره جستجو، این الگوریتم حداقل انحراف بلوک سراسری را که با بهترین بلوک منطبق متناظر است، بدست می دهد. دقت خوب و یافتن

^۱ Full Search Algorithm

بهترین تطبیق از مزایای جستجوی کامل است اما میزان محاسبات آن زیاد می باشد. به همین دلیل استفاده از الگوریتم های تطبیق بلوک سریع به منظور کاهش هزینه محاسبات ضروری است.



شکل ۳-۲: تخمین حرکت جستجوی کامل [۱۸]

۳-۳ الگوریتم جستجوی سه مرحله ای (TSS)

الگوریتم جستجوی سه مرحله ای^۱ (TSS) ابتدا توسط کوا^۲ و همکارانش در سال ۱۹۸۱ معرفی شد [۲۰]. این الگوریتم به دلیل سادگی و عملکرد نزدیک به بهینه، برای کاربردهای ویدئویی با نرخ بیت کم بسیار مورد استفاده قرار گرفت. TSS از الگوی جستجوی بزرگ به کوچک^۳ برای یافتن بهترین بردار حرکت استفاده می کند، که در شکل ۳-۳ نشان داده شده است. مراحل الگوریتم بصورت زیر شرح داده شده است:

گام ۱: اندازه گام اولیه انتخاب می شود. هشت بلوک به فاصله اندازه گام از مرکز (حول بلوک مرکزی) به منظور مقایسه بررسی می شوند.

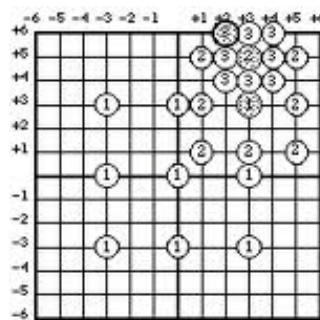
گام ۲: اندازه گام نصف می شود. مرکز به نقطه با حداقل انحراف منتقل می شود.

^۱Three Step Search

^۲Koga

^۳Course to fine

اکثراً اندازه گام ۴ در نظر گرفته می شود و الگوریتم زمانیکه اندازه گام کوچکتر از ۱ شود، خاتمه می یابد. نقطه ای که کوچکترین مقدار معیار انحراف (BDM) را در میان همه نقاط بررسی شده دارا باشد، بعنوان بردار حرکت نهایی انتخاب می شود [۱۹]. تعداد نقاط بررسی شده در این الگوریتم ۲۵ است. ممکن است TSS کمینه سراسری معیار تطبیق را پیدا نکند و در عوض تنها کمینه محلی را بیابد؛ این امر کیفیت سیستم جبران حرکت را کاهش می دهد. الگوریتم TSS از یک الگوی جستجوی توزیع شده بصورت یکنواخت^۱ استفاده می کند، که برای مشخص کردن حرکت های کوچک ظاهر شده در بلوک های ثابت یا نیمه ثابت کارآمد نیست [۲۱ و ۲۲]. برای رفع این مشکل، فنون دیگری پیشنهاد شده اند که جستجو را برای مقیاس های نامعلوم حرکت قابل قبول تر می کند.



شکل ۳-۳: الگوریتم جستجوی سه مرحله ای

۳-۴ الگوریتم جستجوی سه مرحله ای جدید (NTSS)

الگوریتم جستجوی سه مرحله ای جدید^۲ (NTSS) برای تخمین حرکت (ME) سریع، توسط آر. لی^۳ و همکارانش در سال ۱۹۹۴ معرفی شد [۲۳]. NTSS از اولین الگوریتم هایی است که بطور گسترده برای پیاده سازی استانداردهایی نظیر MPEG-1 و H.261 [۲۴] مورد

^۱Uniformly allocated search pattern

^۲New Three Step Search

^۳R. Li

استفاده قرار گرفت. الگوی جستجو در هر گام ثابت است و این الگوریتم هیچ عملیات آستانه‌گذاری را شامل نمی‌شود. با این وجود، در کاربردهای ویدئویی با آهنگ بیت کم به خوبی عمل می‌کند. الگوریتم NTSS علاوه بر اینکه سادگی و باقاعدگی روش TSS را حفظ می‌کند، بر حسب خطای جبران حرکت^۱ و کارآمدی بهتر از TSS عمل کرده، و بر حسب پیچیدگی محاسباتی با آن کاملاً سازگار است.

NTSS با TSS از دو جهت متفاوت است: (۱) فرض الگوی نقاط مورد بررسی متمرکز شده در مرکز^۲ در گام اول؛ (۲) تلفیق روش توقف در نیمه مسیر^۳ برای بلوک‌های ثابت یا نیمه ثابت [۱۹].

شکل ۳-۴ روش جستجوی NTSS را نشان می‌دهد. در گام اول، الگوریتم NTSS از الگوی جستجوی شبکه‌ای 3×3 در مرکز همراه با شبکه بزرگتر 9×9 در TSS استفاده می‌کند. اگر نقطه با حداقل BDM در مرکز پنجره جستجو باشد، جستجو متوقف می‌شود؛ اگر نقطه با حداقل BDM یکی از نقاط روی شبکه 3×3 باشد، تنها پنج یا سه نقطه دیگر بررسی می‌شود؛ در غیر اینصورت مرکز پنجره جستجو به نقطه با حداقل BDM حرکت کرده و ادامه روش مشابه TSS می‌باشد. جزئیات این روش بصورت زیر است.

گام ۱: روی هم‌رفته $17 = 9 + 8$ نقطه شامل ۹ نقطه مرکزی روی شبکه 3×3 و ۸ نقطه همسایه روی شبکه 9×9 بررسی می‌شود. اگر نقطه با حداقل BDM در مرکز پنجره جستجو باشد، جستجو خاتمه می‌یابد؛ در غیر اینصورت به گام ۲ برو.

گام ۲: اگر یکی از هشت نقطه همسایه روی شبکه 3×3 ، نقطه با حداقل BDM در گام اول باشد، به گام ۳ برو؛ در غیر اینصورت به گام ۴ برو.

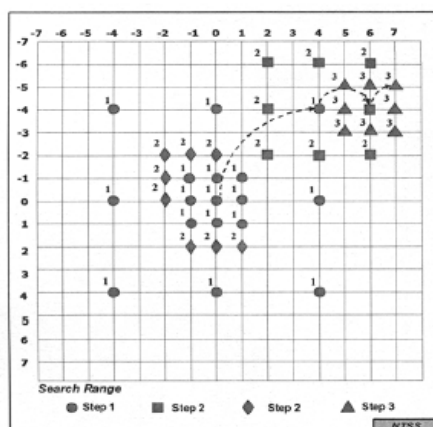
^۱ Motion compensation error

^۲ Center-biased checking point pattern

^۳ Halfway-stop

گام ۳: مرکز پنجره جستجوی 3×3 کوچک به نقطه برنده بدست آمده از گام اول حرکت می‌کند. بر طبق موقعیت نقطه برنده قبلی پنج یا سه نقطه دیگر بررسی می‌شوند، سپس جستجو تمام می‌شود.

گام ۴: اندازه پنجره جستجوی 9×9 بزرگ به میزان زیادی کوچک شده و مرکز آن به نقطه با حداقل BDM بدست آمده از گام ۱ حرکت می‌کند، ادامه جستجو مشابه TSS است.



شکل ۳-۴: الگوریتم جستجوی سه مرحله‌ای جدید

اگرچه NTSS از نقاط بررسی بیشتری در گام اول استفاده می‌کند، اما توقف در گام اول و گام دوم بطور قابل توجهی محاسبات را کاهش می‌دهد. با توجه به روش توقف در نیمه مسیر، NTSS به ۱۷ نقطه جستجو برای بلوک‌های ثابت و ۲۰ یا ۲۲ نقطه برای بردارهای حرکت کوچک داخل ناحیه 5×5 مرکزی نیاز دارد. در بدترین حالت (یعنی بدون بلوک ثابت)، $33 = 8 + 25$ نقطه جستجو در مقایسه با ۲۵ نقطه در TSS مورد نیاز است [۲۵ و ۲۶].

۳-۵ الگوریتم جستجوی چهار مرحله ای (FSS)

الگوریتم جستجوی چهار مرحله ای^۱ (FSS) در سال ۱۹۹۶ توسط لای-مان پو^۲ و وینگ-چان ما^۳ پیشنهاد شد [۲۷]. این الگوریتم از الگوی جستجوی متمرکز شده در مرکز با ۹ نقطه بررسی روی پنجره ۵×۵ استفاده می کند. FSS عملکرد بهتری نسبت به TSS و عملکردی مشابه با NTSS دارد. این الگوریتم در بدترین حالت به ۲۷ نقطه بررسی به جای ۳۳ نقطه NTSS نیاز دارد. الگوریتم FSS بصورت زیر خلاصه شده است.

گام ۱: نقطه با حداقل انحراف بلوک (BDM) از میان ۹ نقطه بررسی روی پنجره ۵×۵ قرار گرفته در مرکز ناحیه جستجوی ۱۵×۱۵ پیدا می شود، که در شکل ۳-۵ (الف) نشان داده شده است. اگر نقطه با حداقل BDM در مرکز پنجره جستجو باشد، به گام ۴ برو؛ در غیر اینصورت به گام ۲ برو.

گام ۲: اندازه پنجره جستجو ۵×۵ باقی می ماند. اما الگوی جستجو به موقعیت نقطه با حداقل BDM قبلی وابسته است.

(الف) اگر نقطه با حداقل BDM قبلی در گوشه پنجره جستجوی قبلی قرار داشته باشد، پنج نقطه دیگر بررسی می شود، که در شکل ۳-۵ (ب) نشان داده شده است.

(ب) اگر نقطه با حداقل BDM قبلی در وسط محور افقی یا عمودی پنجره جستجوی قبلی قرار داشته باشد، سه نقطه دیگر بررسی می شود، که در شکل ۳-۵ (ج) نشان داده شده است.

اگر نقطه با حداقل BDM بدست آمده در این گام، در مرکز پنجره جستجو قرار داشته باشد، به گام ۴ برو؛ در غیر اینصورت به گام ۳ برو.

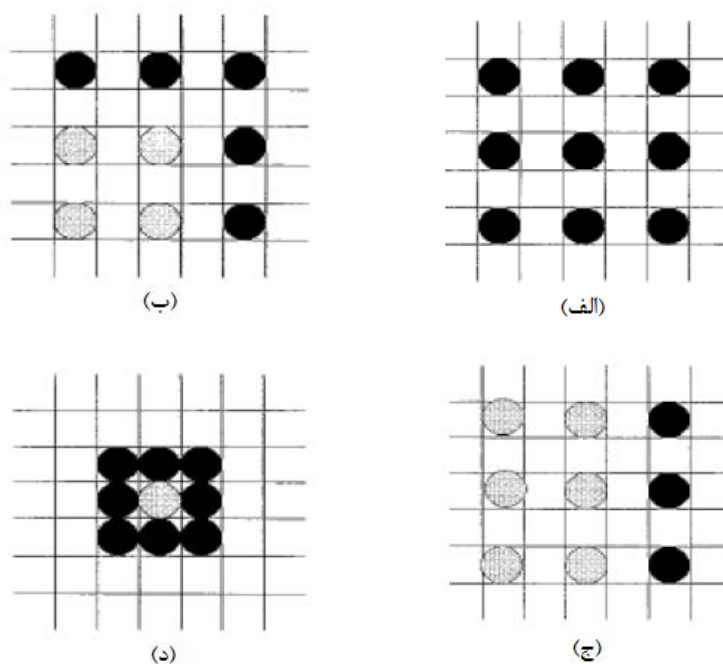
^۱Four Step Search

^۲Lai-Man Po

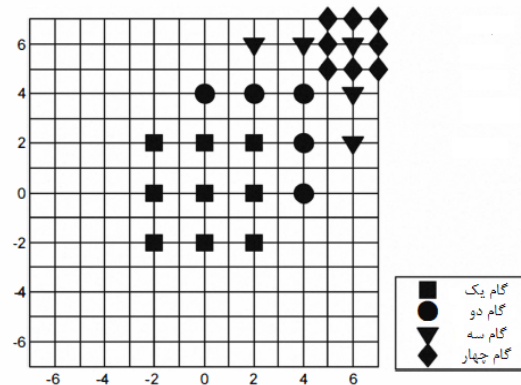
^۳Wing-Chan Ma

گام ۳: استراتژی الگوی جستجو مشابه با گام ۲ است، اما در نهایت به گام ۴ می‌رود.
 گام ۴: پنجره جستجو به 3×3 کاهش می‌یابد، که در شکل ۳-۵ (د) نشان داده شده است.
 است. جهت بردار حرکت نهایی با توجه به نقطه با حداقل BDM بدست آمده از میان این ۹ نقطه جستجو مشخص می‌شود.

شکل ۳-۶ پروسه الگوریتم FSS را نشان می‌دهد. با توجه به الگوریتم، ممکن است از گام‌های میانی FSS پرش شده و مستقیماً به گام نهایی با پنجره 3×3 بروید، اگر نقطه با حداقل BDM در مرکز پنجره جستجو قرار بگیرد. بر اساس این الگوی جستجوی چهار مرحله‌ای، می‌توان تمام جابجایی پنجره 15×15 و حتی پنجره‌های کوچکتر 5×5 و 3×3 را پوشش داد.



شکل ۳-۵: الگوی جستجوی FSS (الف) گام اول، (ب) گام دوم/سوم، (ج) گام دوم/سوم، (د) گام چهارم [۱۹]



شکل ۳-۶: پروسه الگوریتم FSS [۲۸]

الگوریتم FSS عملکرد بهتری نسبت به TSS و عملکردی مشابه با NTSS دارد. در بدترین حالت، تعداد نقاط مورد بررسی در FSS ۲۷ می باشد که تنها ۲ نقطه بیشتر از TSS است [۲۸ و ۱۹].

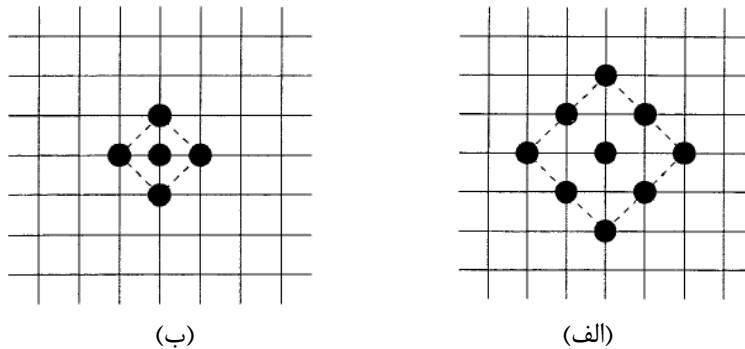
۳-۶ الگوریتم جستجوی لوزی (DS)

الگوریتم جستجوی لوزی^۱ (DS) از دو الگوی جستجو استفاده می کند [۲۹]. الگوی اول، الگوی جستجوی لوزی بزرگ^۲ (LDSP) نامیده شده و شامل ۹ نقطه بررسی، ۱ نقطه مرکزی و ۸ نقطه حول مرکز به شکل لوزی، می باشد، که در شکل ۳-۷ (الف) نشان داده شده است. الگوی دوم شامل ۵ نقطه بررسی به شکل لوزی کوچکتر می باشد، که الگوی جستجوی لوزی کوچک^۳ (SDSP) نامیده می شود و در شکل ۳-۷ (ب) نشان داده شده است.

^۱Diamond Search

^۲Large Diamond Search Pattern

^۳Small Diamond Search Pattern



شکل ۳-۷: (الف) الگوی جستجوی لوزی بزرگ (LDSP)، (ب) الگوی جستجوی لوزی کوچک (SDSP) [۲۹]

در پروسه جستجوی الگوریتم DS، LDSP بصورت تکرار شونده استفاده می شود تا زمانی که حداقل انحراف بلوک^۱ در نقطه مرکزی اتفاق بیفتد. سپس الگوی جستجو از LDSP به SDSP، بعنوان آخرین مرحله جستجو، تغییر می کند. از میان ۵ نقطه بررسی در SDSP، نقطه ای که حداقل انحراف بلوک را داشته باشد، بردار حرکت بهترین بلوک منطبق را مشخص می کند.

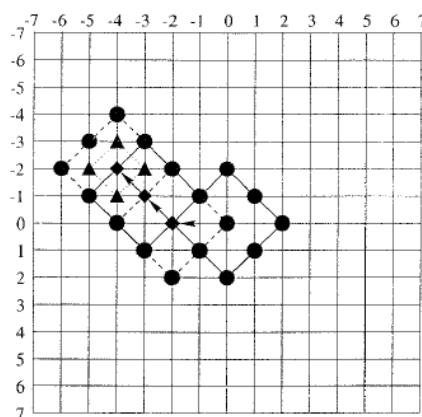
الگوریتم DS بصورت زیر خلاصه شده است.

گام ۱: LDSP اولیه در مرکز پنجره جستجو قرار می گیرد، و ۹ نقطه LDSP بررسی می شود. اگر نقطه با حداقل انحراف بلوک محاسبه شده در مرکز قرار بگیرد، به گام ۳ برو؛ در غیر این صورت، به گام ۲ برو.

گام ۲: نقطه با حداقل انحراف بلوک بدست آمده از گام قبل بعنوان نقطه مرکزی LDSP جدید در نظر گرفته می شود. اگر نقطه با حداقل انحراف بلوک جدید در مرکز باشد، به گام ۳ برو؛ در غیر این صورت، این گام را تکرار کن.

^۱Minimum block distortion

شکل ۳-۸ پروسه الگوریتم DS را نشان می‌دهد. در این الگوریتم نقاط بررسی بین گام‌های متوالی بصورت جزئی هم‌پوشانی دارند؛ به ویژه، هنگامیکه LDSP بصورت تکرار شونده استفاده شود. اگر نقطه با حداقل انحراف بلوک قبلی در یکی از گوشه‌ها یا لبه LDSP قرار بگیرد، به ترتیب تنها پنج یا سه نقطه برای بررسی مورد نیاز است. اگر نقطه مرکزی LDSP، حداقل انحراف بلوک را دارا باشد، الگوی جستجو به SDSP تغییر کرده و تنها چهار نقطه جدید بررسی می‌شود.



الگوریتم DS برای رشته‌های تصویر در محدوده وسیع جابجایی‌های حرکت^۱ به خوبی عمل می‌کند. عملکرد این الگوریتم بهتر از TSS و نزدیک به NTSS می‌باشد، درحالی‌که محاسبات را تقریباً ۲۰٪-۲۵٪ کاهش می‌دهد [۳۰ و ۱۹].

५७

۳-۷ الگوریتم جستجوی الگوی صلیبی و فقی (ARPS)

۳-۷-۱ نمایش از الگوریتم

سرعت و دقت الگوریتم های جستجوی مبتنی بر الگو، به اندازه الگوی جستجو و دامنه بردار حرکت ^۱ (MV) هدف وابسته است. بنابراین استفاده از الگوهای جستجوی مختلف بر اساس رفتار حرکت تخمین زده شده (بر حسب دامنه حرکت) برای بلوک فعلی مطلوب است [۳۱]. در اینجا دو سؤال مطرح می شود: (۱) چگونه می توان رفتار حرکت بلوک فعلی را به منظور انجام تخمین حرکت مناسب، از پیش تعیین کرد؟ (۲) چه اندازه و شکلی برای الگوهای جستجو مناسب است؟

در خصوص سؤال اول، در بیشتر موارد، ماکروبلوک های ^۲ (MBs) مجاور متعلق به یک شیء متحرک، حرکت های مشابه ای دارند. بنابراین، رفتار حرکتی بلوک فعلی می تواند با توجه به بردارهای حرکت بلوک های همسایه اش در حوزه های فضایی یا زمانی پیشگویی شود. با توجه به سؤال دوم، از دو نوع الگوی جستجو استفاده می شود. الگوی اول، الگوی صلیبی و فقی ^۳ (ARP) با بازوی صلیبی قابل تنظیم (اندازه الگو) است، که برای هر ماکروبلوک بر طبق رفتار حرکتی پیشگویی شده اش تعیین می شود. ARP تنها یکبار در آغاز جستجوی هر ماکروبلوک بکار برده می شود. هدف یافتن نقطه شروع خوب برای مرحله جستجوی محلی است تا از جستجوی میانی غیرضروری جلوگیری شده و خطر به تله افتادن در کمینه محلی برای مسیر جستجوی طولانی کاهش یابد. نقطه آغاز جدید تا حد ممکن به کمینه سراسری نزدیک است. اگر چنین باشد، یک الگوی جستجوی کوچک، فشرده و با اندازه ثابت قادر خواهد بود به سرعت جستجوی محلی باقیمانده را کامل کند. این الگوی جستجوی کوچک

^۱ Motion Vector

^۲ MacroBlock

^۳ Adaptive Rood Pattern

بصورت تکرار شونده و نامحدود در جستجوی محلی استفاده می شود تا زمانی که بردار حرکت نهایی بدست آید.

۳-۷-۲ پیشگویی بردار حرکت هدف

به منظور دستیابی به پیشگویی^۱ دقیق بردار حرکت بلوک فعلی، دو عامل باید در نظر گرفته شود: (۱) انتخاب ناحیه پشتیبانی^۲ (ROS) که شامل بلوک های همسایه ای است که بردار حرکت آن ها در محاسبه بردار حرکت پیشگویی شده استفاده می شود، و (۲) الگوریتم مورد استفاده برای محاسبه بردار حرکت پیشگویی شده.

در حوزه زمان، بلوک در فریم مرجع که در موقعیت مشابه با فریم فعلی قرار گرفته، یک انتخاب مستقیم برای ROS زمانی می باشد. بعلاوه، بلوک های همسایه ای فریم مرجع نیز می توانند نامزدهای خوبی برای پیشگویی باشند. با این حال، استفاده از همبستگی زمانی^۳ نیاز به ذخیره کردن بردارهای حرکت قبلی دارد، که در پیاده سازی های عملی با فضای ذخیره سازی محدود نامطلوب است. بنابراین، در این روش تنها از همبستگی فضایی^۴ استفاده می شود.

از آنجایی که در حوزه فضایی همه بلوک های درون فریم ویدئو بصورت افقی پویش^۵ می شوند، تنها بردارهای حرکت بلوک های مجاور چپ، بالا، بالا-چپ و بالا-راست برای بلوک فعلی بعنوان ROS در دسترس می باشد. استفاده از بلوک های بیشتر پیچیدگی محاسبات را افزایش می دهد، به همین دلیل ROS فضایی به بلوک های همسایه ذکر شده محدود می شود که در شکل ۳-۹ نشان داده شده است. نوع الف هر چهار بلوک همسایه را پوشش می دهد،

^۱Prediction

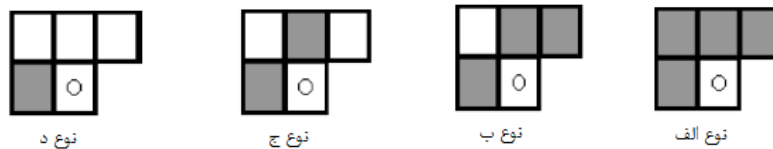
^۲Region Of Support

^۳Temporal correlation

^۴Spatial correlation

^۵Raster-scan

نوع ب ROS پذیرفته شده در تعدادی از استانداردهای بین المللی مانند H.263 [۲۴] است. نوع ج شامل دو بلوک مجاور در جهات اصلی است، و نوع د تنها شامل بلوک مجاور در سمت چپ بلوک فعلی می باشد.



شکل ۳-۹: چهار نوع ROS مشخص شده با بلوک های تیره. بلوک با علامت "○" بلوک فعلی است [۳۱].

یک روش رایج برای بدست آوردن بردار حرکت پیشگویی شده، محاسبه میانگین آماری بردارهای حرکت در ROS می باشد.

آزمایشات بسیاری با چهار نوع ROS و دو ضابطه پیشگویی، میانگین و میانه، انجام شده است. نتایج این آزمایشات نشان می دهد که ROS ها و ضابطه های پیشگویی ذکر شده عملکردی مشابه بر حسب PSNR (با اختلاف ۰.۱ dB) و تعداد نهایی نقاط بررسی مورد نیاز (با اختلاف کمتر از ۵٪) دارند. بنابراین، در این روش ساده ترین ROS (یعنی نوع د) استفاده می شود، که به حداقل حافظه نیاز دارد زیرا تنها یک بردار حرکت همسایه را ذخیره می کند [۳۱].

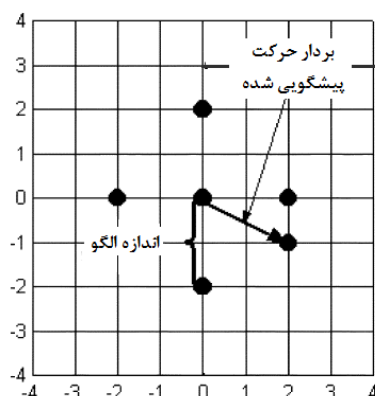
۳-۷-۳ انتخاب الگوهای جستجو

در این روش از دو نوع الگوی جستجو استفاده می شود:

(۱) الگوی افقی - برای جستجوی اولیه: شکل الگو، صلیبی متقارن است و چهار نقطه جستجو در چهار رأس آن قرار دارد، که در شکل ۳-۱۰ نشان داده شده است. ساختار اصلی ARP به شکل صلیبی است و اندازه آن به فاصله بین هر رأس و نقطه مرکزی اشاره دارد. به دلیل متقارن بودن، چهار نقطه روی رأس فاصله یکسانی تا مرکز دارند. در حالت خاص،

هنگامیکه اندازه ARP صفر باشد، از شکل صلیبی معمولی به نقطه مرکزی تغییر می‌کند. انتخاب شکل صلیبی اولاً بر اساس مشاهده ویژگی حرکت رشته‌های ویدئو در دنیای واقعی است. لازم به ذکر است که توزیع بردار حرکت در جهات افقی و عمودی بیشتر از جهات دیگر است [۳۲]، زیرا اکثر جابجایی‌های دوربین در این جهات می‌باشد. علاوه بر اینکه شکل صلیبی الگو همه جهات افقی و عمودی را شامل می‌شود، ARP می‌تواند به سرعت چنین حرکت‌هایی را آشکار کرده و جستجو مستقیماً به ناحیه محلی کمینه سراسری پرش کند. دوماً، هر بردار حرکت می‌تواند به مؤلفه‌های افقی و عمودی تجزیه شود. الگو با شکل صلیب برای هر شیء متحرک، که ممکن است بردار حرکتی در هر جهت داشته باشد، حداقل می‌تواند گرایش عمده جهت حرکت شیء را آشکار کند که نتیجه مطلوبی در مرحله جستجوی اولیه است. بعلاوه، تقارن در شکل ARP نه تنها برای پیاده‌سازی سخت‌افزاری مزیت است، بلکه کارآمدی آن را افزایش می‌دهد.

علاوه بر الگوی صلیبی با چهار بازو، اضافه کردن بردار حرکت پیشگویی شده به ARP مطلوب است، زیرا احتمال زیادی وجود دارد که این بردار حرکت مشابه با بردار حرکت هدف باشد. به این ترتیب، احتمال آشکار کردن حرکت دقیق در مرحله اولیه افزایش می‌یابد.



شکل ۳-۱۰: الگوی صلیبی افقی (ARP) [۳۱]

در الگوریتم ARP، طول چهار بازوی الگوی صلیبی برابر است. طول بازوی، Γ ، این الگو بر اساس بردار حرکت بلوک همسایه چپ بلوک فعلی بصورت زیر بدست می آید.

$$\Gamma = Round|\overrightarrow{MV}_{predicted}|$$

$$= Round\left[\sqrt{MV_{predicted}^2(x) + MV_{predicted}^2(y)}\right] \quad (1-3)$$

که $MV_{predicted}^2(x)$ و $MV_{predicted}^2(y)$ به ترتیب مؤلفه های افقی و عمودی بردار حرکت پیشگویی شده می باشند. عملگر روند^۱ نزدیک ترین عدد صحیح آرگومان را می گیرد. عملگر Γ تعریف شده در معادله (۱-۳) شامل عملیات جذر و مجذور است، که دشواری پیاده سازی سخت افزاری را افزایش می دهد. برای رفع این مشکل، می توان تنها از یکی از دو مؤلفه بردار حرکت پیشگویی شده، که مقدار مطلق بزرگتری دارد، بعنوان اندازه ARP در نظر گرفت.

$$\Gamma = Max\{|\overrightarrow{MV}_{predicted}(x)|, |\overrightarrow{MV}_{predicted}(y)|\} \quad (2-3)$$

از نظر ریاضی، دامنه مؤلفه بردار حرکت با مقدار مطلق بزرگتر به طول بردار حرکت اصلی نزدیک تر است، و بنابراین معادله (۲-۳) تقریب خوبی برای دامنه حرکت می باشد. در این الگوریتم از معادله (۲-۳) به منظور تعیین اندازه ARP استفاده می شود.

توجه شود که ROS انتخاب شده (نوع د) برای بلوک های سمت چپ در هر فریم قابل اجرا نیست. برای آن بلوک ها، از بردارهای حرکت همسایه استفاده نمی شود، و طول بازو با اندازه ثابت ۲ پیکسل (یعنی $\Gamma=2$) برای ARP پذیرفته می شود. هم چنین، اندازه های بازوی بزرگتر در نظر گرفته نمی شود، زیرا معمولاً ماکروبلوک های مرزی در فریم به پس زمینه ثابت تعلق دارند و احتمال کمتری وجود دارد که حرکت های بزرگ داشته باشند.

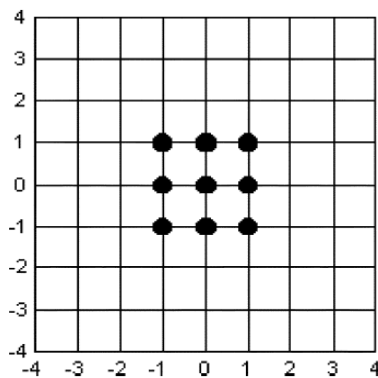
^۱ Round

بطور خلاصه، اگر بردار حرکت پیشگویی شده صفر نباشد، الگوریتم ARP شامل ۵ (بدون هم‌پوشانی) یا ۴ (با هم‌پوشانی) نقطه جستجو در مرحله جستجوی اولیه است؛ در غیر اینصورت، تنها یک نقطه جستجو مورد نیاز است.

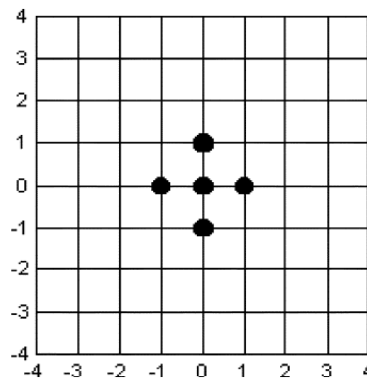
(۲) الگوی ثابت- برای جستجوی محلی: در مرحله جستجوی اولیه با استفاده از ARP مرکز جستجوی جدیدی بدست می‌آید، که حول کمینه سراسری است. در مرحله جستجوی محلی، به جای استفاده از FS یا سایر BMA های دیگر می‌توان از الگوی جستجوی کوچک، فشرده و ثابت بصورت نامحدود به منظور یافتن کمینه سراسری استفاده کرد. هنگامیکه الگوی ثابت استفاده می‌شود، نقطه با حداقل خطای تطبیق^۱ (MME) بدست آمده در گام فعلی، بعنوان مرکز جستجوی تکرار بعدی در نظر گرفته می‌شود. این روند ادامه می‌یابد تا زمانی که نقطه MME در مرکز الگوی ثابت قرار بگیرد. دو نوع الگوی جستجوی فشرده پیشنهاد شده است: (۱) الگوی صلیبی با اندازه واحد^۲ (URP)، که مشابه SDSP در الگوریتم DS است (شکل ۱۱-۳ الف)؛ (۲) الگوی مربعی ۳×۳ (شکل ۱۱-۳ ب). الگوی مربعی ۳×۳ در مقایسه با URP، PSNR مشابه‌ای دارد اما از نقاط جستجوی بیشتری (۴۰٪-۸۰٪) استفاده می‌کند. از این‌رو، در الگوریتم ARP از URP در مرحله جستجوی محلی استفاده می‌شود.

^۱Minimal Matching Error

^۲Unit-size Rood Pattern



(ب)



(الف)

شکل ۳-۱۱: دو الگوی جستجوی ثابت، الف) URP ب) الگوی مربعی [۳۱]

۳-۷-۴ پیش‌قضاوت حرکت صفر (ZMP)

در بسیاری از کاربردهای مخابرات ویدئویی از قبیل ویدئو تلفن، حرکت کوچکی بین فریم‌های مجاور وجود دارد. از این‌رو، درصد بالایی از بلوک‌های بدون حرکت^۱ در این قبیل رشته‌های ویدئویی وجود دارد. بنابراین، اگر از پیش‌قضاوت حرکت صفر^۲ (ZMP) در آغاز تخمین حرکت استفاده شود، محاسبات به میزان قابل توجهی کاهش می‌یابد. بررسی‌های بیشتر مشخص کرده است که متوسط خطاهای تطبیق^۳ بلوک‌های ثابت بسیار کمتر از بلوک‌های متحرک است. بنابراین، با محاسبه خطای تطبیق بین بلوک فعلی و بلوک در موقعیت مشابه در فریم مرجع و مقایسه آن با آستانه از پیش تعیین شده، T، می‌توان این پیش‌قضاوت را ایجاد کرد. اگر خطای تطبیق کمتر از T باشد، بلوک فعلی ثابت در نظر گرفته شده و جستجو خاتمه می‌یابد.

^۱Zero-motion block^۲Zero-Motion Prejudgment^۳Matching error

معمولا در تخمین حرکت اندازه ماکروبلوک 16×16 در نظر گرفته شده و از مجموع قدرمطلق اختلاف (SAD) بعنوان معیار خطای تطبیق استفاده می شود. با اعمال FS روی ۹ رشته ویدئو با نرخ بیت های رمزگذاری متفاوت، مشخص شد که متوسط SAD ماکروبلوک های ثابت در محدوده ۶۰۰-۱۳۰۰ است [۳۱]. از آنجایی که آستانه بالاتر خطای پیشگویی بزرگتری دارد، در این روش $T=512$ انتخاب می شود.

برای رشته های ویدئویی با جابجایی های بزرگ روش ZMP کارایی ندارد، زیرا درصد بلوک های ثابت در این رشته ها بسیار کم است.

توجه شود که آستانه بسته به نوع کاربرد، قابل تنظیم است. برای مثال، اگر کیفیت ویدئو مهم نباشد، این آستانه بزرگتر انتخاب می شود تا سرعت جستجو بیشتر گردد.

۳-۷-۵ خلاصه روش ARPS-ZMP

بلوک همسایه چپ بلوک فعلی (یعنی نوع د) بعنوان ROS برای ARPS-ZMP در نظر گرفته می شود. برای بلوک های مرزی سمت چپ در هر فریم، اندازه ARP ۲ پیکسل (یعنی $\Gamma=2$)، و آستانه پیشنهاد شده $T=512$ است. روش ARPS-ZMP بصورت زیر خلاصه شده است [۳۰ و ۳۳].

گام ۱: محاسبه خطای تطبیق (مرکز SAD) بین بلوک فعلی و بلوک در موقعیت مشابه در فریم مرجع (یعنی مرکز پنجره جستجو). اگر مرکز SAD از T کمتر باشد، بردار حرکت صفر است و جستجو خاتمه می یابد؛ در غیر این صورت، برای بلوک های سمت چپ $\Gamma=2$ و برای سایر بلوک ها Γ از رابطه (۲-۳) بدست می آید.

گام ۲: قرار دادن مرکز ARP در مرکز پنجره جستجو و بررسی چهار نقطه آن به همراه موقعیت بردار حرکت پیشگویی شده، به منظور یافتن نقطه MME فعلی.

گام ۳: قرار دادن مرکز الگوی صلیبی با اندازه واحد (URP) روی نقطه MME بدست آمده از گام قبل و بررسی نقاط آن. اگر نقطه MME جدید در مرکز URP فعلی قرار نگیرد، این گام تکرار می شود؛ در غیر این صورت، بردار حرکت با توجه به نقطه MME مشخص شده در این گام بدست می آید.

۳-۸ الگوریتم تطبیق بلوک به روش پیش بینی متوسط (MPBM)

پایه الگوریتم تطبیق بلوک به روش پیش بینی متوسط^۱ (MPBM)، الگوریتم جستجوی صلیبی افقی (ARPS) است، که در آن اندازه بازو بر اساس متوسط دو بردار حرکت ماکروبلوک های همسایه بدست می آید [۳۴]. این روش از پیشگویی ماکروبلوک های همسایه چپ و بالا استفاده می کند. دو نوع الگوی جستجوی متفاوت، الگوی جستجوی لوزی بزرگ (LDSP) و الگوی جستجوی لوزی کوچک (SDSP)، در این روش استفاده می شود. بعلاوه، MPBM از یک مقدار آستانه از پیش تعیین شده برای خطای بین ماکروبلوک فعلی و ماکروبلوک منطبق بدست آمده از LDSP بهره می گیرد؛ اگر خطا کمتر از مقدار آستانه بود نیازی به SDSP نمی باشد، و بنابراین محاسبات کاهش می یابد. هم چنین، این الگوریتم، از روش جدیدی استفاده می کند که زمان پردازش را با توقف محاسبه مجموع قدرمطلق اختلاف بین پیکسل های باقیمانده در ماکروبلوک فعلی حداقل می کند؛ به این صورت که اگر مقدار مجموع قدرمطلق اختلاف بین ماکروبلوک فعلی و هر ماکروبلوک در LDSP از مقدار

^۱Mean Predictive Block Matching

مجموع قدرمطلق قبلی بیشتر شود، محاسبات متوقف شده و جستجو به موقعیت بعدی در LDSP می‌رود. گام‌های الگوریتم پیشنهاد شده بصورت زیر خلاصه شده است.

گام ۱: محاسبه مجموع قدرمطلق اختلاف (مرکز SAD) بین ماکروبلوک فعلی و ماکروبلوک در موقعیت مشابه در فریم مرجع (یعنی مرکز پنجره جستجو). اگر مجموع قدرمطلق اختلاف (مرکز SAD) کمتر از آستانه از پیش تعیین شده باشد، به این معنا است که حرکتی صورت نگرفته و پردازش خاتمه می‌یابد.

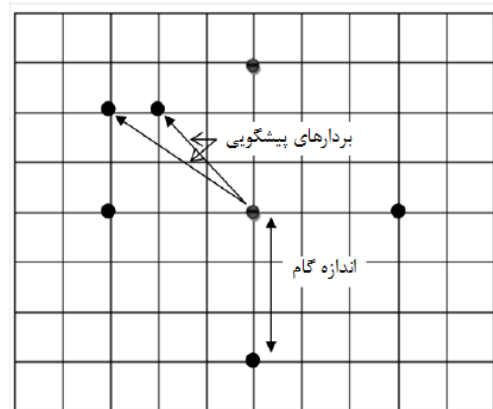
گام ۲: اگر ماکروبلوک در گوشه بالایی چپ باشد، تنها پنج نقطه LDSP جستجو می‌شود، در غیر اینصورت بردار حرکت بالا (MV_A) و بردار حرکت چپ (MV_L) برای پیشگویی اندازه گام بصورت زیر مورد استفاده قرار می‌گیرد.

$$L_x = \text{round}\left(\text{abs}\left(\frac{MV_A(x) + MV_L(x)}{2}\right)\right) \quad (3-3)$$

$$L_y = \text{round}\left(\text{abs}\left(\frac{MV_A(y) + MV_L(y)}{2}\right)\right) \quad (4-3)$$

اندازه گام برابر $\max\{L_x, L_y\}$ است.

گام ۳: سپس از میان نقاط LDSP $\{(0, 0), (0, \pm \text{اندازه گام}), (\pm \text{اندازه گام}, 0)\}$ و بردارهای $\{(MV_A), (MV_L)\}$ ماکروبلوک منطبق جستجو می‌شود، که در شکل ۳-۱۲ نشان داده شده است. اگر مقدار SAD فعلی از مقدار SAD قبلی بیشتر شود آنگاه محاسبات متوقف شده و به موقعیت بعدی پرش می‌کند، در غیر اینصورت همه پیکسل‌ها بررسی می‌شوند. به این ترتیب ماکروبلوک منطبق برای LDSP بدست می‌آید.



شکل ۳-۱۲: الگوی جستجو در مرحله جستجوی اولیه، نقاط توپر جستجوی گام اول در MPBM هستند که شامل الگوی جستجوی لوزی بزرگ (LDSP) و دو بردار حرکت پیشگویی می باشد [۳۴].

گام ۴: اگر خطای ماکروبلوک منطبق در گام ۳ کمتر از مقدار آستانه از پیش تعیین شده باشد آنگاه پردازش متوقف شده و موقعیت ماکروبلوک منطبق، بردار حرکت را مشخص می کند. در غیر اینصورت، موقعیت ماکروبلوک منطبق در گام ۳ مرکز جستجوی جدید شده و چهار نقطه SDSP $\{(0, \pm 1), (\pm 1, 0)\}$ بررسی می شود. اگر ماکروبلوک منطبق در مرکز باقی بماند، محاسبات تمام می شود، در غیر اینصورت پردازش مشابهی تکرار می شود تا زمانی که ماکروبلوک منطبق در مرکز قرار بگیرد. موقعیت ماکروبلوک منطبق بدست آمده بردار حرکت را مشخص می کند.

فصل چهارم

روش پیمناای

۴ فصل چهارم

۴-۱ مقدمه

در فصل ۳ تعدادی از الگوریتم‌های تطبیق بلوک تخمین حرکت مورد بررسی قرار گرفت. در این فصل دو الگوریتم برای تطبیق بلوک پیشنهاد کرده‌ایم که کیفیت تصویر بازسازی شده را بهبود داده و زمان محاسبات کمتری را به خود اختصاص می‌دهد.

۱. تطبیق بلوک به روش پیش‌بینی متوسط بهبود یافته.
۲. الگوریتم جستجوی ساده و کارآمد مبتنی بر جستجوی سه مرحله‌ای.

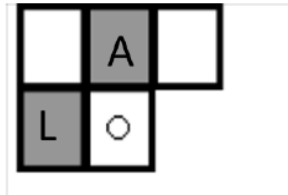
۴-۲ تطبیق بلوک به روش پیش‌بینی متوسط بهبود یافته

در این بخش، الگوریتم جدیدی به نام تطبیق بلوک به روش پیش‌بینی متوسط بهبود یافته^۱ (EMPBM) پیشنهاد می‌کنیم که در واحد تخمین حرکت رمزگذار/رمزگشای تصاویر ویدئویی کاربرد دارد. پایه این الگوریتم، جستجوی الگوی صلیبی افقی است.

همانطور که قبلاً بیان گردید، به دلیل اینکه استفاده از همبستگی زمانی فریم‌ها در یک رشته ویدئو نیاز به ذخیره کردن بردارهای حرکت قبلی داشته و مستلزم فضای ذخیره‌سازی بیشتری است، بنابراین در این الگوریتم تنها از همبستگی فضایی استفاده می‌کنیم. از آنجایی که هر شیء متحرک چندین ماکروبلوک را به خود اختصاص می‌دهد، احتمال زیادی وجود دارد که بردار حرکت ماکروبلوک فعلی با بردار حرکت ماکروبلوک‌های همسایه‌اش مشابه باشد. بنابراین، می‌توان از بردار حرکت ماکروبلوک‌های همسایه برای پیشگویی بردار حرکت

^۱Enhanced Mean Predictive Block Matching

ماکروبلاک فعلی استفاده کرد. این الگوریتم از بردارهای حرکت ماکروبلاک‌های همسایه بالا و چپ به عنوان ROS استفاده می‌کند، که در شکل ۴-۱ نشان داده شده است.



شکل ۴-۱: موقعیت ماکروبلاک‌های پیشگویی [۳۴]

این الگوریتم شامل دو مرحله جستجوی متوالی است: (۱) جستجوی اولیه^۱ و (۲) جستجوی محلی^۲. جستجوی اولیه برای هر ماکروبلاک تنها یکبار در آغاز جستجو بکار می‌رود تا نقطه شروع خوب را برای مرحله جستجوی محلی پیدا کند. با انجام جستجوی اولیه، خطر به تله افتادن در نقاط کمینه محلی خطای تطبیق در حالت جستجوی طولانی به میزان زیادی کاهش می‌یابد. در این الگوریتم برای مرحله جستجوی اولیه، الگوی صلیبی افقی را پیشنهاد کرده‌ایم. شکل الگو صلیبی متقارن است و چهار نقطه جستجو در چهار رأس آن قرار دارد. طول بازوی الگو به فاصله بین هر رأس و نقطه مرکزی اشاره دارد. طول بازوهای افقی و عمودی باهم برابر نمی‌باشد و بر اساس بردارهای حرکت ماکروبلاک‌های همسایه بالا و چپ بصورت زیر تعیین می‌شوند:

$$L_x = \text{round}\left(\left|\frac{MV_a(x) + MV_l(x)}{2}\right|\right) \quad (۱-۴)$$

$$L_y = \text{round}\left(\left|\frac{MV_a(y) + MV_l(y)}{2}\right|\right) \quad (۲-۴)$$

^۱Initial search

^۲Local search

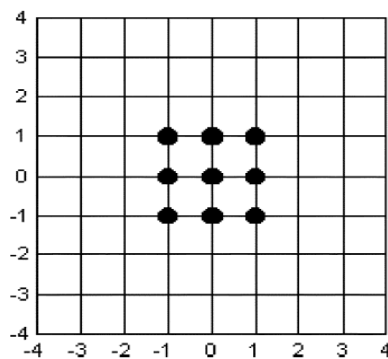
که L_x و L_y به ترتیب طول بازوهای افقی و عمودی و MV_a و MV_l به ترتیب بردارهای حرکت ماکروبلوک‌های همسایه بالا و چپ می‌باشند.

علاوه بر نقاط این الگو، دو موقعیت بدست آمده از بردارهای حرکت ماکروبلوک‌های همسایه بالا و چپ نیز در این مرحله بررسی می‌شوند، زیرا احتمال زیادی وجود دارد که بردار حرکت ماکروبلوک فعلی مشابه بردار حرکت ماکروبلوک‌های همسایه‌اش باشد.

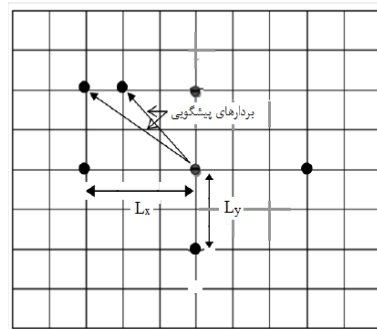
به دلیل اینکه ROS برای ماکروبلوک گوشه بالای چپ (اولین ماکروبلوک در فریم) مصداقی ندارد، برای آن ماکروبلوک تمام نقاط الگوی صلیبی افقی با طول ثابت $L=2$ بررسی می‌شود. در حالت خاص، اگر طول بازو صفر باشد، تنها نقطه مرکزی الگوی صلیبی به همراه دو موقعیت حاصل از دو بردار حرکت پیشگویی شده بررسی می‌شود.

در مرحله جستجوی محلی از الگوی مربعی 3×3 بصورت تکرار شونده و نامحدود استفاده می‌کنیم تا زمانی که بردار حرکت نهایی بدست آید. نقطه با حداقل خطای تطبیق (MME) بدست آمده از مرحله جستجوی اولیه به عنوان مرکز این الگو در نظر گرفته شده و تمام نقاط الگو بررسی می‌شوند. نقطه MME جدید به عنوان مرکز الگو در تکرار بعدی استفاده می‌شود. این روند تا زمانی ادامه می‌یابد که نقطه MME در مرکز الگوی ثابت قرار گیرد و به این ترتیب بردار حرکت نهایی با توجه به موقعیت نقطه MME مشخص می‌شود. شکل ۲-۴ الگوهای مورد استفاده در مراحل جستجوی اولیه و جستجوی محلی را نشان می‌دهد.

به دلیل اینکه اکثر استانداردهای رمزگذاری ویدئو از اندازه بلوک 16×16 استفاده می‌کنند [۱]، ما نیز در الگوریتم پیشنهادی از این اندازه بلوک استفاده می‌نماییم.



(ب)



(الف)

شکل ۴-۲: الگوهای جستجو در مراحل جستجوی اولیه و محلی، الف) جستجوی مرحله اول که در آن از الگوی صلیبی و دو بردار پیشگویی استفاده شده است، ب) الگوی مربعی ۳×۳

همانطور که در فصل قبل بیان گردید، به دلیل اینکه در کاربردهای مخابرات ویدئویی مانند ویدئو تلفن، حرکت کوچکی بین فریم‌های مجاور صورت می‌گیرد، درصد بالایی از ماکروبلوک‌های بدون حرکت در رشته‌های ویدئویی وجود دارد. بنابراین، در الگوریتم پیشنهادی روش پیش‌قضاوت حرکت صفر (ZMP) را بکار برده‌ایم تا از انجام محاسبات اضافی برای ماکروبلوک‌های ثابت بپرهیزیم. استفاده از روش پیش‌قضاوت حرکت صفر محاسبات را به ویژه در رشته‌های ویدئویی با جابجایی کوچک، به میزان زیادی کاهش می‌دهد، اما PSNR را کمی خراب می‌کند. در این تکنیک به منظور تشخیص ماکروبلوک‌های ثابت، آستانه‌ای برای خطای تطبیق بین ماکروبلوک‌ها از پیش تعیین می‌شود. با توجه به مرجع [۳۱] مقدار آستانه برای ماکروبلوک‌های ثابت در محدوده ۶۰۰-۱۳۰۰ می‌باشد. مقدار آستانه بزرگتر سرعت الگوریتم را افزایش داده، اما کیفیت تصویر بازسازی شده را کاهش می‌دهد. در این الگوریتم مقدار آستانه را ۲۵۶ در نظر می‌گیریم زیرا کیفیت تصویر مهم است.

با توجه به مرجع [۱]، معیار خطای تطبیق SAD کمترین پیچیدگی محاسباتی را داشته و کیفیت پیشگویی خوبی دارد. بنابراین در این الگوریتم از SAD به عنوان تابع تطبیق استفاده می‌کنیم.

$$SAD = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \quad (3-4)$$

که C_{ij} و R_{ij} به ترتیب پیکسل‌های فریم فعلی و فریم مرجع و M و N به ترتیب تعداد سطرها و ستون‌های فریم می‌باشند.

به منظور ارزیابی کیفیت ویدئوی بازسازی شده، معمولاً از معیار متوسط مجذور خطا^۱ (MSE) استفاده می‌شود و به صورت زیر تعریف می‌گردد.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2 \quad (4-4)$$

معیار MSE معمولاً در رابطه نسبت اوج سیگنال به نویز (PSNR) بکار می‌رود:

$$PSNR = 10 \log_{10} \left[\frac{225^2}{MSE} \right] \quad (5-4)$$

که ۲۵۵ حداکثر مقدار روشنایی برای رزولوشن ۸ بیت است. معیار PSNR به دلیل سادگی، یک انتخاب بسیار رایج در انجمن رمزگذاری ویدئو^۲ می‌باشد. برای ارزیابی عملکرد الگوریتم پیشنهادی از PSNR استفاده می‌نماییم.

در ادامه گام‌های الگوریتم پیشنهاد شده توضیح داده می‌شود.

گام ۱: آستانه را ۲۵۶ تعیین می‌کنیم. مرکز SAD را بین ماکروبلوک در فریم فعلی و ماکروبلوک در موقعیت مشابه در فریم مرجع (مرکز پنجره جستجو) محاسبه می‌کنیم. اگر

^۱Mean Squared Error

^۲Video coding community

مرکز SAD از آستانه از پیش تعیین شده کوچکتر باشد، به این معناست که حرکتی صورت نگرفته و ماکروبلوک ثابت می‌باشد و جستجو خاتمه می‌یابد؛ در غیر اینصورت، گام ۲ اجرا می‌شود.

گام ۲: طول بازوهای الگوی صلیبی را با توجه به بردار حرکت ماکروبلوک‌های همسایه بالا و چپ و از روابط (۱-۴) و (۲-۴) محاسبه می‌کنیم.

گام ۳: مرکز الگوی صلیبی را در مرکز پنجره جستجو قرار داده و به منظور پیدا کردن موقعیت با حداقل خطای تطبیق (MME) در فریم مرجع، معیار SAD را برای چهار رأس الگو و نیز دو موقعیت بدست آمده از بردارهای حرکت ماکروبلوک‌های همسایه بالا و چپ محاسبه می‌کنیم.

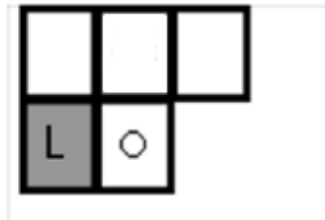
گام ۴: مرکز الگوی مربعی 3×3 را روی نقطه MME بدست آمده از گام قبل قرار داده و معیار SAD را برای نقاط آن، $\{(1, \pm 1), (\pm 1, 1), (0, \pm 1), (\pm 1, 0), (0, 0), (\pm 1, 0), (0, \pm 1)\}$ محاسبه می‌کنیم. اگر نقطه MME جدید در مرکز الگوی مربعی قرار نگرفت، این گام تکرار می‌شود؛ در غیر اینصورت، بردار حرکت نهایی با توجه به موقعیت نقطه MME جدید در این گام تعیین می‌شود.

۳-۴ الگوریتم جستجوی ساده و کارآمد مبتنی بر جستجوی سه

مرحله‌ای

در این بخش، الگوریتم دیگری پیشنهاد می‌کنیم که پایه آن الگوریتم جستجوی سه مرحله‌ای است و در واحد تخمین حرکت رمزگذار/رمزگشای تصاویر ویدئویی کاربرد دارد.

در الگوریتم پیشنهادی از همبستگی فضایی فریم‌ها استفاده کرده و به منظور پیشگویی بردار حرکت ماکروبلوک فعلی تنها از بردار حرکت ماکروبلوک همسایه چپ بهره می‌گیریم، که در شکل ۳-۴ نشان داده شده است.



شکل ۳-۴: موقعیت ماکروبلوک پیشگویی

این الگوریتم شامل دو مرحله جستجوی متوالی می‌باشد: (۱) جستجوی اولیه و (۲) جستجوی محلی. جستجوی اولیه برای هر ماکروبلوک تنها یکبار در آغاز جستجو بکار می‌رود تا نقطه شروع خوب را برای مرحله جستجوی محلی پیدا کند. در الگوریتم پیشنهادی، برای مرحله جستجوی اولیه از الگوی مربعی مورد استفاده در الگوریتم جستجوی سه مرحله‌ای با کمی تغییر استفاده می‌کنیم. در این الگو، اندازه گام‌های افقی و عمودی با هم برابر نیستند و بر اساس مؤلفه‌های افقی و عمودی بردار حرکت ماکروبلوک همسایه چپ بصورت زیر تعیین می‌شوند.

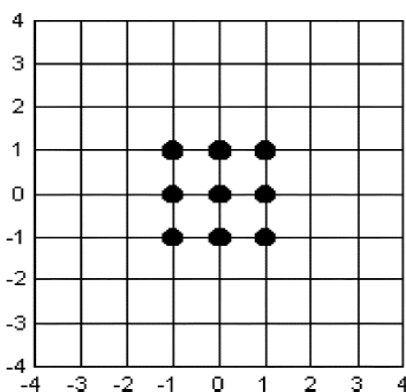
$$|MV_l(x)| = \text{اندازه گام افقی} \quad (۴-۶)$$

$$|MV_l(y)| = \text{اندازه گام عمودی} \quad (۴-۷)$$

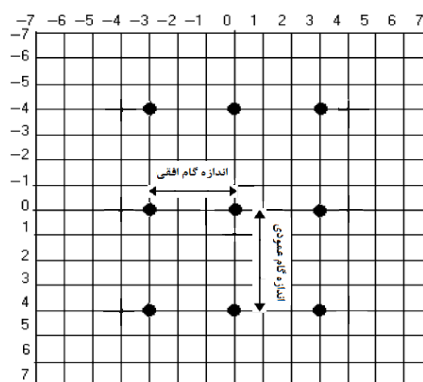
که $MV_l(x)$ و $MV_l(y)$ به ترتیب مؤلفه‌های افقی و عمودی بردار حرکت ماکروبلوک همسایه چپ می‌باشند.

در این مرحله، نقاط الگوی مربعی، که موقعیت آن‌ها بر اساس اندازه گام‌های افقی و عمودی تعیین می‌شود، بررسی می‌شوند. به دلیل اینکه ROS برای ماکروبلوک‌های مرزی چپ مصداقی ندارد، برای آن ماکروبلوک‌ها اندازه گام‌های افقی و عمودی را برابر ۲ در نظر گرفته می‌شود. در حالت خاص، اگر اندازه گام افقی یا عمودی صفر شود، اندازه گام ۱ در نظر گرفته شده و جستجو انجام می‌شود.

در مرحله جستجوی محلی از الگوی مربعی 3×3 بصورت تکرار شونده و نامحدود استفاده می‌شود. نقطه MME بدست آمده از مرحله جستجوی اولیه در مرکز این الگو قرار گرفته و تمام نقاط الگو بررسی می‌شوند. نقطه MME جدید به عنوان مرکز الگو در تکرار بعدی مورد استفاده قرار می‌گیرد. این روند تا زمانی ادامه می‌یابد که نقطه MME در مرکز الگوی ثابت قرار گیرد، و به این ترتیب بردار حرکت نهایی با توجه به موقعیت نقطه MME مشخص می‌شود. شکل ۴-۴ الگوهای مورد استفاده در مراحل جستجوی اولیه و محلی را نشان می‌دهد.



(ب)



(الف)

شکل ۴-۴: الگوهای جستجو در مراحل جستجوی اولیه و محلی، الف) جستجوی مرحله اول که در آن از الگوی مربعی با اندازه گام‌های نامساوی استفاده شده است، ب) الگوی مربعی 3×3

در الگوریتم پیشنهاد شده از اندازه بلوک 16×16 استفاده می‌کنیم. به منظور افزایش سرعت الگوریتم، روش پیش‌قضاوت حرکت صفر (ZMP) را با آن تلفیق می‌نماییم. به دلیل اینکه کیفیت تصویر بازسازی شده مهم است، آستانه را ۲۵۶ در نظر می‌گیریم. در این الگوریتم از معیار SAD به عنوان تابع تطبیق و از نسبت اوج سیگنال به نویز (PSNR) برای ارزیابی کیفیت ویدئوی بازسازی شده استفاده می‌کنیم.

در ادامه گام‌های الگوریتم پیشنهاد شده توضیح داده می‌شود.

گام ۱: آستانه را ۲۵۶ تعیین می‌کنیم. مرکز SAD را بین ماکروبلوک در فریم فعلی و ماکروبلوک در موقعیت مشابه در فریم مرجع (مرکز پنجره جستجو) محاسبه می‌کنیم. اگر مرکز SAD از آستانه از پیش تعیین شده کوچکتر باشد، به این معناست که حرکتی صورت نگرفته و ماکروبلوک ثابت می‌باشد و جستجو خاتمه می‌یابد؛ در غیر اینصورت، گام ۲ اجرا می‌شود.

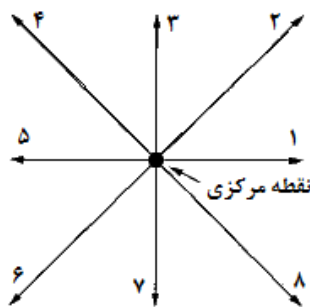
گام ۲: اندازه گام‌های افقی و عمودی را با توجه به بردار حرکت ماکروبلوک همسایه چپ و از روابط $(4-6)$ و $(4-7)$ بدست می‌آوریم.

گام ۳: مرکز الگوی مربعی مورد استفاده در مرحله جستجوی اولیه را در مرکز پنجره جستجو قرار داده و معیار SAD را به منظور یافتن نقطه MME در فریم مرجع، برای نقاط آن محاسبه می‌کنیم.

گام ۴: مرکز الگوی مربعی 3×3 را روی نقطه MME بدست آمده از گام قبل قرار داده و معیار SAD را برای نقاط آن، $\{(1, \pm 1), (\pm 1, 1), (0, \pm 1), (\pm 1, 0), (\pm 1, 0), (0, \pm 1)\}$ محاسبه می‌کنیم. اگر نقطه

MME جدید در مرکز الگوی مربعی قرار نگرفت، این گام تکرار می‌شود؛ در غیر اینصورت، بردار حرکت نهایی با توجه به موقعیت نقطه MME جدید در این گام تعیین می‌شود.

به منظور کاهش تعداد نقاط مورد بررسی در این الگوریتم، الگوی جستجوی دیگری را برای مرحله جستجوی اولیه پیشنهاد داده‌ایم. در الگوی مربعی مورد استفاده علاوه بر نقطه مرکزی، هشت نقطه بررسی وجود دارد، که هشت جهت را برای بردار حرکت تعریف می‌کند. این هشت جهت در شکل ۴-۵ نشان داده شده است.



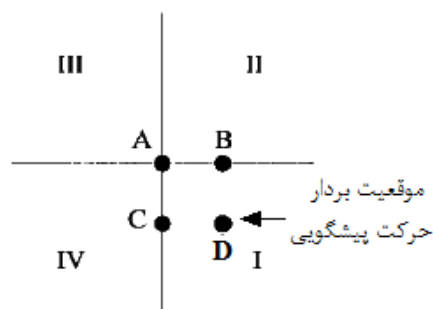
شکل ۴-۵: جهت‌های جستجو در الگوی جستجو [۲۱]

به دلیل تقارن الگو، برای هر جهت، جهت مخالف آن نیز وجود دارد و با توجه به فرض سطح خطای تک‌قله‌ای^{۱*} بررسی تعدادی از این جهت‌ها غیرضروری است. برای پرهیز از بررسی نقاط غیرضروری، باید محدوده‌ای را پیدا کنیم که کمینه سراسری خطای تطبیق در آن قرار داشته باشد. از اینرو، جستجو را به دو فاز تقسیم می‌کنیم. در فاز اول، صفحه جستجو را به چهار ناحیه تقسیم کرده و نقطه مرکزی، دو نقطه در جهت افقی و عمودی، و نقطه حاصل از بردار حرکت پیشگویی شده را بررسی می‌کنیم که در شکل ۴-۶ نشان داده شده است. در فاز دوم، بر اساس مقدار SAD بدست آمده از بررسی این نقاط، می‌توان ناحیه

^۱Unimodal error surface assumption

* سطح خطای تک قله‌ای دارای این ویژگی است که هر چه از موقعیت کمینه خطای سراسری دورتر شویم، خطای تطبیق بلوک بصورت یکنوا افزایش می‌یابد.

بردار حرکت ماکروبلوک فعلی را تشخیص داده و تعداد نهایی نقاط مورد بررسی در این مرحله را تعیین کنیم. با یک مثال الگوی پیشنهادی را توضیح می‌دهیم. همانطور که در شکل ۴-۶ نشان داده شده است، اگر بردار حرکت ماکروبلوک همسایه چپ در یک چهارم اول قرار بگیرد، SAD برای نقاط A، B، C و D محاسبه می‌شود. در این الگو نقاط A، B و C همیشه ثابت هستند اما نقطه D باتوجه به بردار حرکت پیشگویی شده مشخص می‌شود.



شکل ۴-۶: چهار ناحیه جستجو

قانونی که برای تعیین محدوده بردار حرکت ماکروبلوک فعلی استفاده می‌کنیم، بصورت زیر است.

اگر $SAD(D) < SAD(A)$ ، $SAD(D) < SAD(B)$ و $SAD(D) < SAD(C)$ باشد، ناحیه I انتخاب می‌شود؛

اگر $SAD(A) \geq SAD(B)$ و $SAD(A) \geq SAD(C)$ باشد، ناحیه I انتخاب می‌شود؛

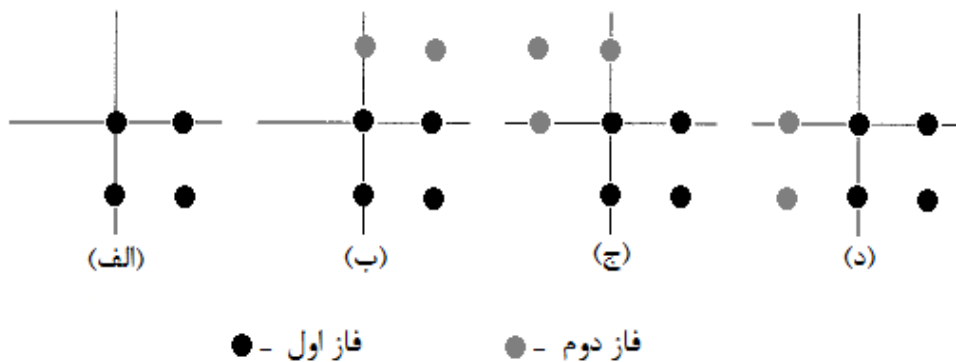
اگر $SAD(A) \geq SAD(B)$ و $SAD(A) < SAD(C)$ باشد، ناحیه II انتخاب می‌شود؛

اگر $SAD(A) < SAD(B)$ و $SAD(A) < SAD(C)$ باشد، ناحیه III انتخاب می‌شود؛

اگر $SAD(A) < SAD(B)$ و $SAD(A) \geq SAD(C)$ باشد، ناحیه IV انتخاب می‌شود.

الگوی کامل بررسی نقاط برای یافتن موقعیت حداقل خطا در شکل ۴-۷ نشان داده شده است.

قانون بیان شده با توجه به موقعیت بردار حرکت پیشگویی شده کمی تغییر می‌کند، اما قاعده بکار رفته در آن ثابت است.



شکل ۴-۷: الگوهای جستجو بر اساس هر یک از یک چهارم‌های انتخاب شده، (الف) یک چهارم I، (ب) یک چهارم II، (ج) یک چهارم III، و (د) یک چهارم IV.

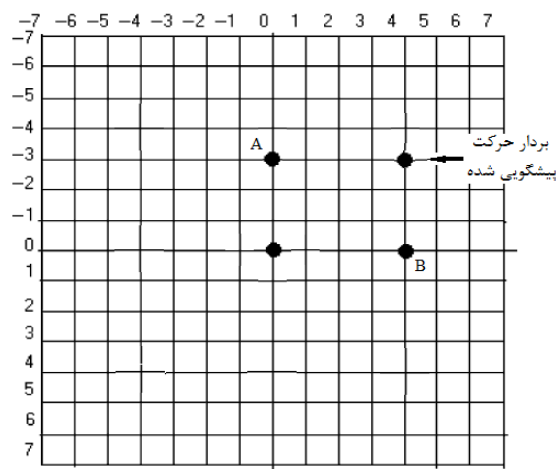
در این الگو، مشابه الگوی اول، اندازه گام برای دو نقطه در جهت افقی و عمودی با توجه به بردار حرکت ماکروبلوک همسایه چپ تعیین می‌شود. در حالت خاص، اگر اندازه گام افقی یا عمودی صفر شود، اندازه گام را ۱ در نظر می‌گیریم.

به دلیل اینکه ROS برای ماکروبلوک‌های مرزی چپ مصداقی ندارد، برای آن ماکروبلوک‌ها همه نقاط الگوی مربعی را با اندازه گام ثابت ۲ بررسی می‌کنیم. نقطه MME بدست آمده از این مرحله، به عنوان مرکز جستجو در مرحله جستجوی محلی بکار می‌رود. ادامه الگوریتم همانند قبل است.

الگوی دیگری نیز در مرحله جستجوی اولیه به منظور کاهش تعداد نقاط جستجو پیشنهاد کرده‌ایم. به دلیل اینکه در کاربردهای با نرخ بیت کم، حرکت کوچکی بین فریم‌های رشته ویدئو صورت می‌گیرد و هم‌چنین هر شیء متحرک چندین ماکروبلوک را به خود

اختصاص می‌دهد، می‌توان الگویی پیشنهاد داد که در آن تنها نقاط اطراف بردار حرکت پیشگویی شده بررسی شود.

در الگوی پیشنهادی، تنها چهار نقطه بررسی می‌شود: نقطه مرکزی، نقطه بدست آمده از بردار حرکت ماکروبلوک همسایه چپ، و دو نقطه در جهت افقی و عمودی که در محدوده بردار حرکت پیشگویی شده قرار دارند. فاصله این دو نقطه‌ی افقی و عمودی تا نقطه مرکزی بر اساس مؤلفه افقی و عمودی بردار حرکت پیشگویی شده تعیین می‌شود. شکل ۴-۸ نمونه‌ای از موقعیت نقاط جستجو را در مرحله جستجوی اولیه نشان می‌دهد. نقطه‌ای که کمترین مقدار SAD را داشته باشد، به عنوان مرکز جستجو در مرحله جستجوی محلی بکار می‌رود. ادامه الگوریتم همانند قبل است.



شکل ۴-۸: نمونه‌ای از موقعیت نقاط جستجو در مرحله جستجوی اولیه، نقاط A و B دو نقطه در محدوده بردار حرکت هستند.

فصل پنجم

نتیج

۵ فصل پنجم

۵-۱ مقدمه

در فصل چهارم به منظور بهبود کیفیت ویدئوی بازسازی شده و افزایش سرعت پردازش، دو الگوریتم پیشنهاد شد. الگوریتم اول بر اساس الگوی جستجوی صلیبی وفقی و الگوریتم دوم بر اساس الگوی جستجوی سه مرحله‌ای عمل می‌کند. پیاده‌سازی سخت‌افزاری این الگوریتم‌ها به دلیل استفاده از الگوهای جستجوی ساده و قابل تنظیم به راحتی امکان‌پذیر است.

در این فصل، مقادیر MSE، PSNR، زمان پردازش و سرعت برای الگوریتم‌های پیشنهادی محاسبه شده و با مقادیر بدست آمده از الگوریتم‌های تطبیق بلوک استاندارد قبلی مقایسه می‌شود. این الگوریتم‌های استاندارد شامل FS، TSS، NTSS، FSS و DS می‌باشند.

شبیه‌سازی با استفاده از نرم‌افزار MATLAB انجام می‌شود. رشته‌های ویدئوی آزمون مورد استفاده در این شبیه‌سازی به سه کلاس تقسیم می‌شوند: کلاس ۱ (جزئیات فضایی کم و میزان حرکت کم)، کلاس ۲ (جزئیات فضایی متوسط و میزان حرکت کم یا برعکس)، و کلاس ۳ (جزئیات فضایی زیاد و میزان حرکت متوسط یا برعکس). رزولوشن این رشته‌ها بصورت QCIF [۱] است. این رزولوشن برای کاربردهای با نرخ بیت بسیار کم استفاده می‌شود.

رشته‌های ویدئوی آزمون عبارتند از: آکیو، کلر^۱، کانتینر^۲ (کلاس ۱)، مادر-دختر^۳، فروشنده^۴ (کلاس ۲)، و سرکارگر، تلفن ماشین^۵، فوتبال^۶، اتوبوس^۷ (کلاس ۳) [۳۵]. ابعاد فریم در همه این رشته‌ها ۱۷۶×۱۴۴ بوده و تنها مؤلفه روشنایی آن‌ها بررسی می‌شود.

۵-۲ نتایج الگوریتم پیشنهادی اول

در این بخش نتایج بدست آمده از اعمال الگوریتم تطبیق بلوک به روش پیش‌بینی متوسط بهبود یافته را با الگوریتم‌های استاندارد ذکر شده در بخش قبل مقایسه می‌کنیم. در این مقایسه ۳۰ فریم اول هر رشته آزمون بررسی می‌شوند. برای هر فریم، فریم قبل از آن به عنوان فریم مرجع در نظر گرفته می‌شود. بردار حرکت برای همه فریم‌ها غیر از فریم اول جستجو می‌شود.

بهترین PSNR را الگوریتم FS دارد و سایر الگوریتم‌ها سعی دارند PSNR نزدیک به FS را ارائه داده، در حالیکه تعداد نقاط جستجو را کاهش دهند. در این شبیه‌سازی، نتایج با توجه به الگوریتم FS ارزیابی می‌شوند.

۵-۲-۱ نتایج کمی

به منظور ارزیابی عملکرد الگوریتم پیشنهاد شده، مقادیر MSE و PSNR را برای ویدئوی‌های بازسازی شده محاسبه کرده و نتایج حاصل را با مقادیر بدست آمده از

^۱ Claire

^۲ Container

^۳ Mother-daughter

^۴ Salesman

^۵ Carphone

^۶ Football

^۷ Bus

الگوریتم‌های بیان شده مقایسه می‌کنیم. جدول‌های ۱-۵ و ۲-۵ به ترتیب مقادیر میانگین

MSE و PSNR را برای ۳۰ فریم اول رشته‌های آزمون نشان می‌دهند.

جدول ۱-۵: میانگین MSE برای ۳۰ فریم اول رشته‌های آزمون

الگوریتم رشته آزمون	FS	TSS	NTSS	FSS	DS	الگوریتم پیشنهادی بدون آستانه‌گذاری	الگوریتم پیشنهادی با آستانه‌گذاری
Akiyo	۴۰.۶۵۲	۴۰.۶۹۱	۴۰.۶۹۱	۴۰.۶۹۱	۴۰.۶۹۱	۴۰.۶۹۱	۴۰.۷۱۵
Claire	۵.۱۳۴۱	۵.۱۵۱۵	۵.۱۳۴۵	۵.۱۵۱۳	۵.۱۳۴۳	۵.۱۳۴۵	۵.۱۸۴۷
Container	۶.۶۸۳۶	۶.۶۸۴۷	۶.۶۸۶۲	۶.۶۸۶۴	۶.۶۸۶۴	۶.۶۸۶۳	۶.۶۹۲۰
Mother-daughter	۳.۴۰۷۱	۳.۴۵۴۵	۳.۴۰۸۸	۳.۴۴۰۴	۳.۴۰۸۰	۳.۴۰۷۵	۳.۴۱۶۲
Salesman	۱۳.۸۱۸۷	۱۳.۸۴۳۳	۱۳.۸۶۷۴	۱۳.۸۳۴۰	۱۳.۸۴۳۳	۱۳.۸۴۳۳	۱۳.۸۴۴۸
Foreman	۳۸.۳۲۲۷	۴۰.۳۴۸۳	۳۹.۵۴۱۸	۴۰.۴۵۳۲	۴۰.۱۶۱۴	۳۹.۱۰۶۱	۳۹.۱۲۵۱
Carphone	۳۹.۸۲۹۱	۴۴.۴۰۳۴	۴۰.۷۳۲۶	۴۴.۵۵۷۱	۴۱.۳۲۱۱	۴۰.۴۳۸۲	۴۰.۳۸۸۴
Football	۷۲۳.۷۷۳۱	۷۵۸.۶۲۱۷	۷۶۴.۶۹۷۹	۸۰۵.۴۵۴۳	۸۵۸.۴۲۶۶	۸۰۵.۳۶۷۱	۸۰۴.۰۹۶۵
Bus	۴۵۸.۸۷۴۷	۵۴۷.۴۳۴۱	۵۸۳.۷۳۹۸	۶۸۲.۱۶۳۳	۶۸۳.۹۴۱۶	۵۱۲.۰۰۶۳	۵۱۱.۵۲۱۵

جدول ۵-۲: میانگین PSNR برای ۳۰ فریم اول رشته‌های آزمون

الگوریتم رشته آزمون	الگوریتم پیشنهادی بدون آستانه‌گذاری	الگوریتم پیشنهادی با آستانه‌گذاری	FS	TSS	NTSS	FSS	DS
Akiyo	۴۴.۶۳۴۵	۴۴.۶۳۳۸	۴۴.۶۳۳۸	۴۴.۶۳۳۸	۴۴.۶۳۳۸	۴۴.۶۳۳۸	۴۴.۶۳۳۸
Claire	۴۱.۸۱۷۲	۴۱.۸۰۸۶	۴۱.۸۱۷۱	۴۱.۸۰۸۲	۴۱.۸۱۵۶	۴۱.۸۱۷۱	۴۱.۷۳۹۱
Container	۴۳.۲۹۶۲	۴۳.۲۹۶۱	۴۳.۲۹۶۰	۴۳.۲۹۶۰	۴۳.۲۹۶۰	۴۳.۲۹۶۰	۴۳.۲۸۶۶
Mother-daughter	۴۴.۷۴۵۳	۴۴.۷۲۰۲	۴۴.۷۴۴۸	۴۴.۷۲۷۸	۴۴.۷۴۴۴	۴۴.۷۴۴۷	۴۴.۷۲۴۴
Salesman	۳۸.۳۸۲۰	۳۸.۳۷۸۴	۳۸.۳۷۵۴	۳۸.۳۸۰۱	۳۸.۳۷۸۴	۳۸.۳۷۸۴	۳۸.۳۷۷۲
Foreman	۳۲.۸۱۴۶	۳۲.۵۹۷۹	۳۲.۶۹۵۴	۳۲.۵۹۸۱	۳۲.۶۰۴۲	۳۲.۷۴۵۱	۳۲.۷۴۳۲
Carphone	۳۲.۴۰۵۷	۳۱.۹۶۳۴	۳۲.۳۲۱۹	۳۱.۹۵۰۷	۳۲.۲۶۱۷	۳۲.۳۴۹۱	۳۲.۳۵۳۸
Football	۱۹.۷۴۱۷	۱۹.۵۲۴۹	۱۹.۴۸۷۸	۱۹.۲۶۵۲	۱۹.۰۰۳۹	۱۹.۲۷۸۹	۱۹.۲۸۶۰
Bus	۲۱.۶۲۴۸	۲۰.۹۰۸۶	۲۰.۶۴۹۲	۱۹.۹۶۴۵	۱۹.۹۶۴۲	۲۱.۱۵۲۵	۲۱.۱۵۱۸

جدول‌های ۳-۵ و ۴-۵ به ترتیب زمان پردازش و سرعت^۱ الگوریتم را برای ۳۰ فریم اول رشته‌های آزمون نشان می‌دهند. سرعت الگوریتم را بصورت نسبت زمان پردازش در الگوریتم FS به زمان پردازش در الگوریتم مورد بررسی تعریف می‌کنیم.

^۱Speed up

جدول ۵-۳: زمان مورد نیاز برای پردازش ۳۰ فریم اول رشته‌های آزمون بر حسب ثانیه

الگوریتم رشته آزمون	FS	TSS	NTSS	FSS	DS	الگوریتم پیشنهادی بدون آستانه‌گذاری	الگوریتم پیشنهادی با آستانه‌گذاری
Akiyo	۴.۷۷۱۹	۲.۵۷۸۹	۱.۳۳۶۰	۱.۵۳۷۱	۱.۳۴۲۸	۱.۳۴۶۶	۰.۳۸۴۴
Claire	۴.۸۵۵۰	۲.۶۳۵۵	۱.۵۲۲۸	۱.۶۵۱۵	۱.۴۸۴۹	۱.۴۶۴۳	۰.۴۲۱۹
Container	۴.۹۱۷۴	۲.۷۵۶۵	۱.۵۱۲۵	۱.۶۷۴۱	۱.۴۹۰۴	۱.۴۵۷۱	۰.۵۳۱۷
Mother-daughter	۴.۷۶۴۱	۲.۶۱۱۸	۱.۴۴۴۰	۱.۵۹۰۰	۱.۴۲۱۰	۱.۳۹۰۲	۰.۴۵۸۱
Salesman	۴.۹۴۶۸	۲.۷۳۲۱	۱.۴۴۱۹	۱.۶۳۴۷	۱.۴۲۳۱	۱.۴۰۳۱	۰.۶۲۷۴
Foreman	۴.۸۱۷۳	۲.۶۳۹۵	۱.۷۸۳۲	۱.۸۱۰۰	۱.۶۸۴۲	۱.۵۶۰۷	۱.۳۵۸۳
Carphone	۴.۹۰۲۷	۲.۶۶۹۴	۱.۸۱۹۳	۱.۸۴۴۵	۱.۷۲۰۵	۱.۶۲۳۳	۱.۲۹۲۲
Football	۴.۹۰۱۰	۲.۶۷۰۴	۲.۴۵۶۷	۲.۵۴۰۴	۲.۵۸۰۱	۲.۲۲۱۲	۲.۱۸۲۱
Bus	۴.۹۳۶۴	۲.۶۷۸۰	۲.۵۳۶۰	۲.۴۲۰۱	۲.۴۶۲۰	۱.۷۴۱۵	۱.۷۱۹۷

جدول ۵-۴: سرعت پردازش

الگوریتم رشته آزمون	FS	TSS	NTSS	FSS	DS	الگوریتم پیشنهادی بدون آستانه گذاری	الگوریتم پیشنهادی با آستانه گذاری
Akiyo	۱	۱.۸۵۰۴	۳.۵۷۱۸	۳.۱۰۴۵	۳.۵۵۳۷	۳.۵۴۳۷	۱۲.۴۱۳۹
Claire	۱	۱.۸۴۲۲	۳.۱۸۸۲	۲.۹۳۹۸	۳.۲۶۹۶	۳.۳۱۵۶	۱۱.۵۰۷۵
Container	۱	۱.۷۸۳۹	۳.۲۵۱۲	۲.۹۳۷۳	۳.۲۹۹۴	۳.۳۷۴۸	۹.۲۴۸۴
Mother-daughter	۱	۱.۸۲۴۱	۳.۲۹۹۲	۲.۹۹۶۳	۳.۳۵۲۶	۳.۴۲۶۹	۱۰.۳۹۹۷
Salesman	۱	۱.۸۱۰۶	۳.۴۳۰۸	۳.۰۲۶۱	۳.۴۷۶۱	۳.۵۲۵۶	۷.۸۸۴۶
Foreman	۱	۱.۸۲۵۱	۲.۷۰۱۵	۲.۶۶۱۵	۲.۸۶۰۳	۳.۰۸۶۶	۳.۵۴۶۶
Carphone	۱	۱.۸۳۶۶	۲.۶۹۴۸	۲.۶۵۸۰	۲.۸۴۹۶	۳.۰۲۰۲	۴.۰۷۸۱
Football	۱	۱.۸۳۵۳	۱.۹۹۵۰	۱.۹۲۹۲	۱.۸۹۹۵	۲.۲۰۶۵	۲.۲۴۶۰
Bus	۱	۱.۸۴۳۳	۱.۹۴۶۵	۲.۰۳۹۸	۲.۰۰۵۰	۲.۸۳۴۶	۲.۸۷۰۵

با توجه به جداول ۵-۱ و ۵-۲ مشاهده می شود مقدار متوسط مجذور خطا (MSE) و PSNR الگوریتم پیشنهادی برای رشته های آزمون متعلق به کلاس های ۱ و ۲، که جابجایی زیادی ندارند، تقریباً مشابه و در مواردی کمی بهتر از الگوریتم های استاندارد قبلی است و نزدیک به الگوریتم FS می باشد و برای رشته های متعلق به کلاس ۳، که جابجایی زیادی دارند، به میزان قابل توجهی بهتر شده است. با توجه به نتایج بدست آمده، همانطور که انتظار داشتیم با اعمال روش ZMP مقدار MSE در الگوریتم پیشنهادی برای رشته های آزمون متعلق به کلاس های ۱ و ۲ افزایش و در نتیجه مقدار PSNR کاهش یافت؛ اما میزان این افت زیاد نیست و می توان از آن چشم پوشی کرد. در رشته های آزمون متعلق به کلاس ۳

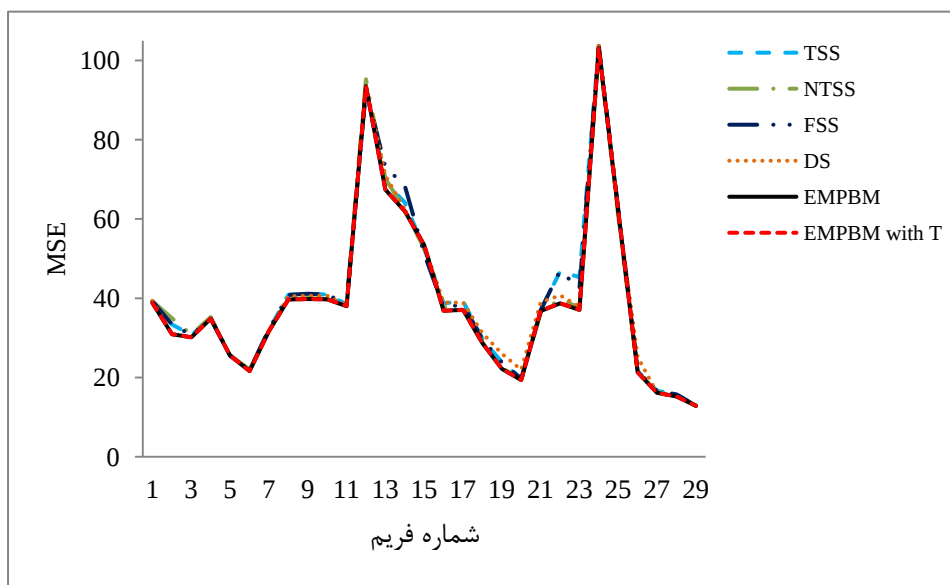
به دلیل جابجایی زیاد، بلوک‌های ثابت کمتری وجود دارد و اعمال روش ZMP تأثیر قابل توجهی در مقادیر MSE و PSNR ندارد و نتایج بدست آمده در اکثر موارد بهتر از الگوریتم‌های قبلی می‌باشد، که این امر در جداول نیز مشاهده می‌شود.

با بررسی جداول ۳-۵ و ۴-۵ ملاحظه می‌شود که زمان پردازش الگوریتم پیشنهادی کمی کمتر از الگوریتم‌های دیگر است و با اعمال روش ZMP این زمان بسیار کاهش می‌یابد. به عبارت دیگر سرعت پردازش الگوریتم بسیار زیاد شده و نسبت به الگوریتم FS برای کلاس ۱، ۱۲.۴۱~۹.۲۵ برابر، کلاس ۲، ۱۰.۴۰~۷.۸۸ برابر و برای کلاس ۳، ۴.۰۹~۲.۲۵ برابر شده است.

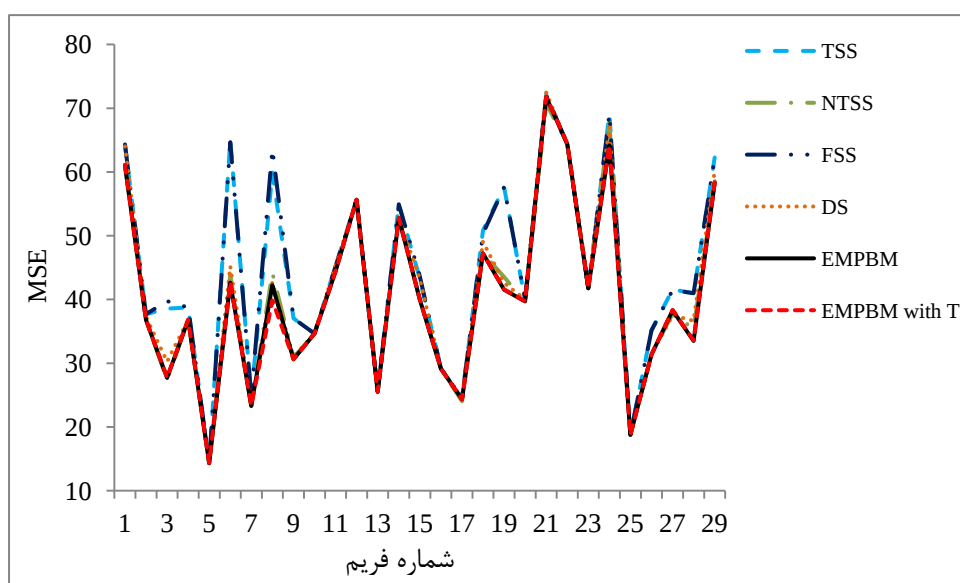
همانطور که نتایج نشان می‌دهند، استفاده از الگوی صلیبی افقی و بردارهای حرکت ماکروبلوک‌های همسایه بالا و چپ در مرحله جستجوی اولیه به خوبی محدوده کمینه سراسری را مشخص می‌کند و استفاده از الگوی مربعی ۳×۳ در مرحله جستجوی محلی با دقت زیادی نقطه MME نهایی را پیدا می‌کند. به این ترتیب، علاوه بر اینکه تعداد نقاط جستجو کاهش و در نتیجه زمان پردازش کاهش می‌یابد، مقدار PSNR افزایش یافته و ویدئو با کیفیت بیشتری بازسازی می‌شود. در کاربردهای مخابرات ویدئو بصورت بلادرنگ سرعت پردازش مهم است و این الگوریتم با توجه به اینکه PSNR خوبی داشته و زمان پردازش بسیار کمتری دارد، می‌تواند در این قبیل کاربردها به خوبی کارایی داشته باشد.

به دلیل اینکه مقادیر MSE و PSNR برای رشته‌های آزمون متعلق به کلاس‌های ۱ و ۲ تقریباً مشابه است، مقایسه عملکرد الگوریتم پیشنهادی با الگوریتم‌های قبلی از طریق رسم نمودار برای این رشته‌های آزمون امکان پذیر نیست. بنابراین، تنها برای رشته‌های آزمون کلاس ۳ این مقایسه را انجام می‌دهیم. شکل‌های ۱-۵ تا ۴-۵ مقایسه‌ای از عملکرد الگوریتم

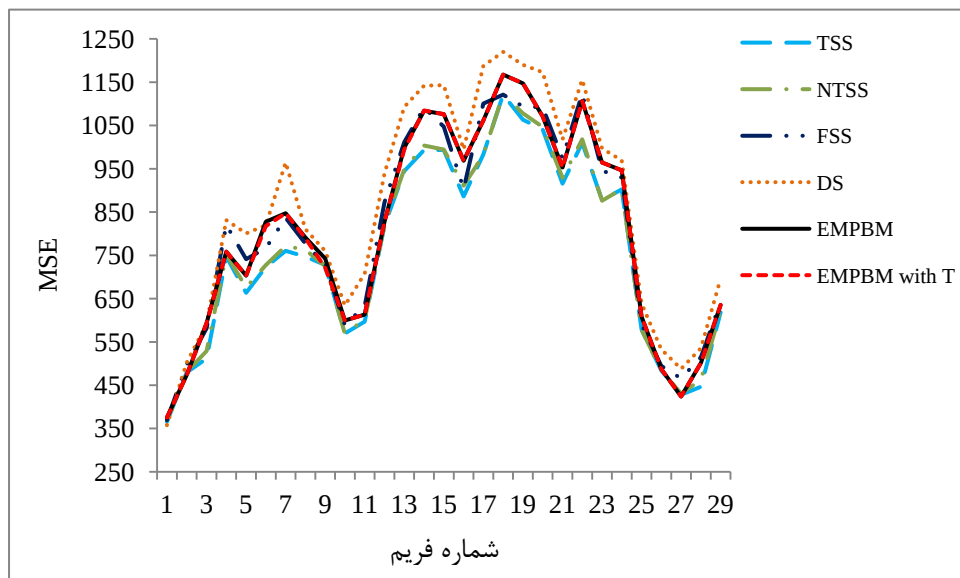
پیشنهادی را با الگوریتم‌های قبلی بر حسب میانگین مجذور خطا (MSE) روی فریم‌های بازسازی شده برای رشته‌های تست سرکارگر، تلفن ماشین، فوتبال و اتوبوس نشان می‌دهند.



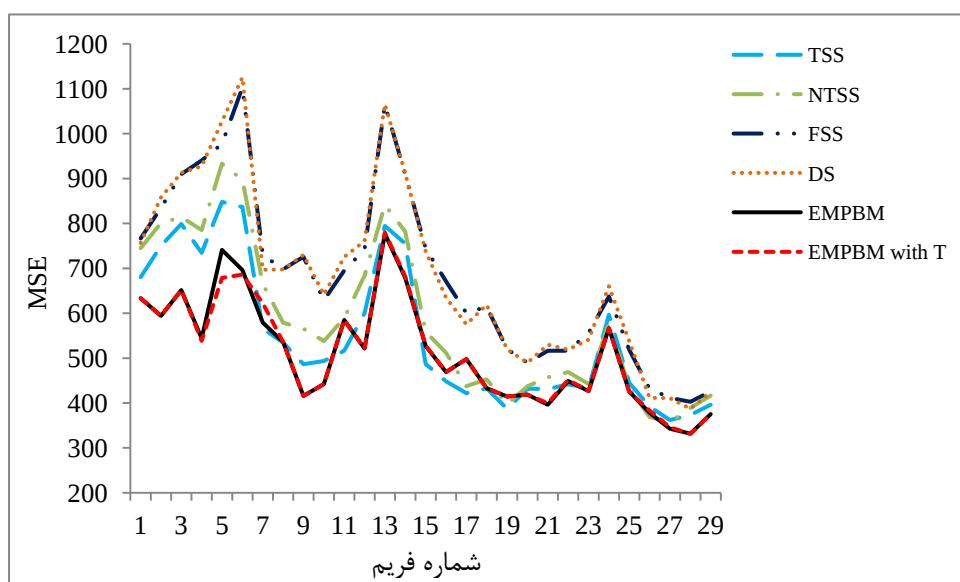
شکل ۵-۱: مقایسه عملکرد بر حسب MSE روی فریم برای سرکارگر



شکل ۵-۲: مقایسه عملکرد بر حسب MSE روی فریم برای تلفن ماشین

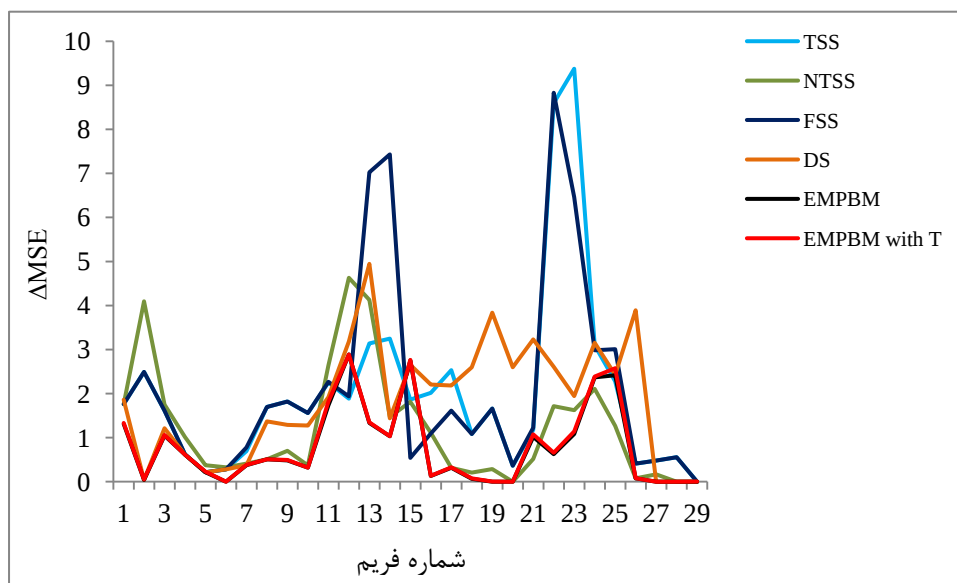


شکل ۳-۵: مقایسه عملکرد بر حسب MSE روی فریم برای فوتبال

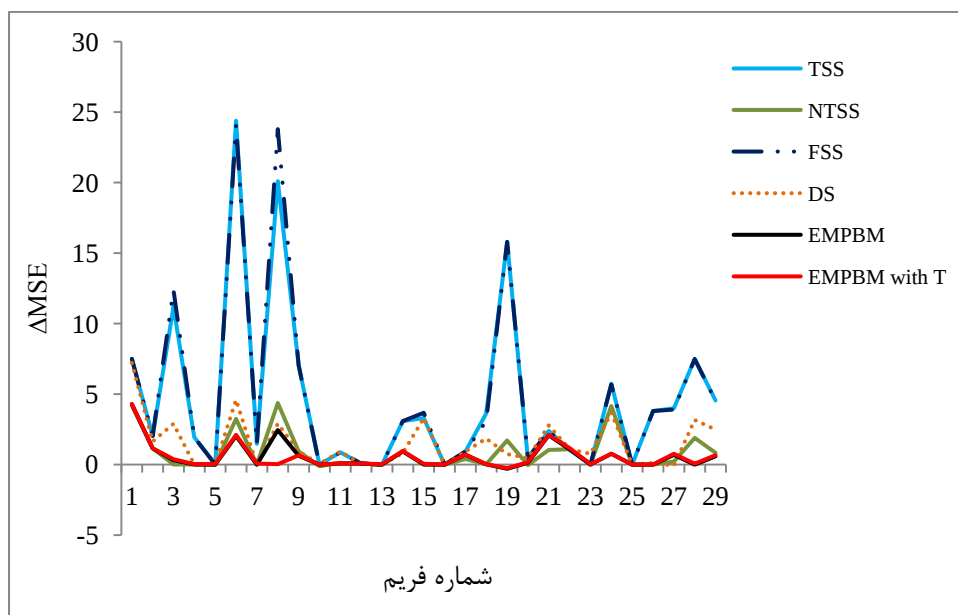


شکل ۴-۵: مقایسه عملکرد بر حسب MSE روی فریم برای اتوبوس

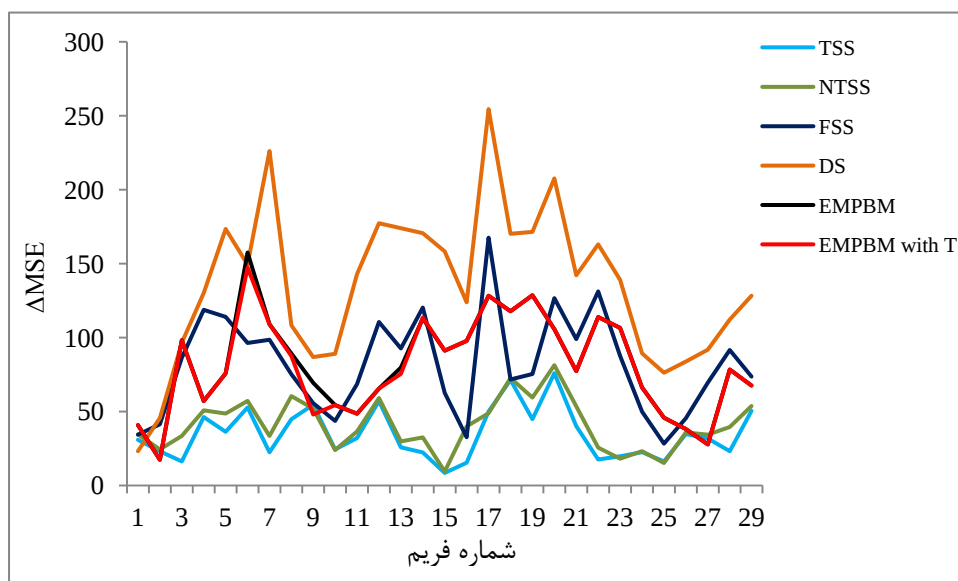
همانطور که قبلاً بیان گردید، بهترین نتیجه مربوط به الگوریتم FS است. از این رو، برای درک بهتر تفاوت میان الگوریتم پیشنهادی و الگوریتم‌های قبلی، اختلاف مقادیر MSE حاصل از همه الگوریتم‌ها را با مقادیر MSE الگوریتم FS بدست آورده و نمودار این اختلافات را رسم می‌کنیم. شکل‌های ۵-۵ تا ۸-۵ نمودار ΔMSE همه الگوریتم‌ها برای رشته‌های آزمون سرکارگر، تلفن ماشین، فوتبال و اتوبوس را نشان می‌دهند.



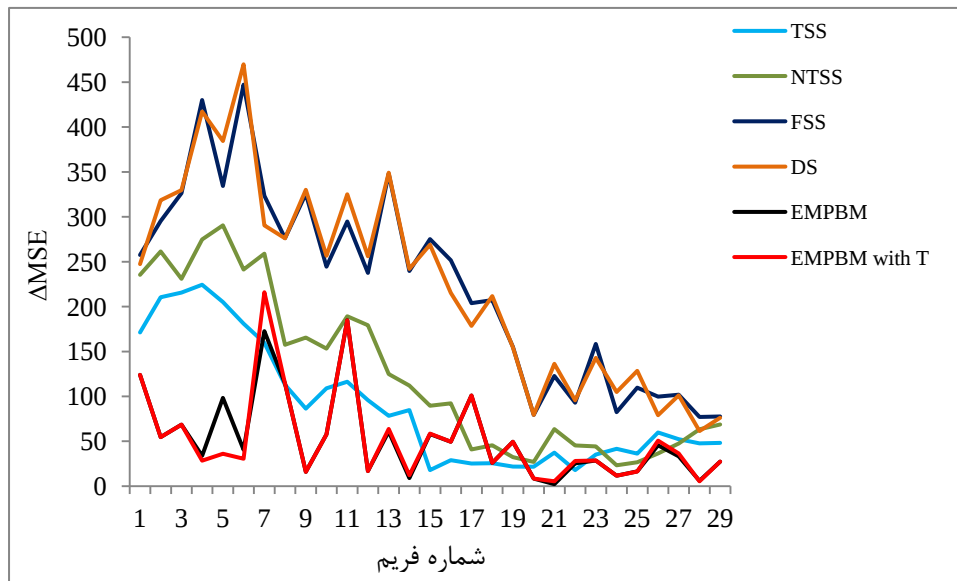
شکل ۵-۵: مقایسه عملکرد بر حسب ΔMSE روی فریم برای سرکارگر



شکل ۵-۶: مقایسه عملکرد بر حسب ΔMSE روی فریم برای تلفن ماشین

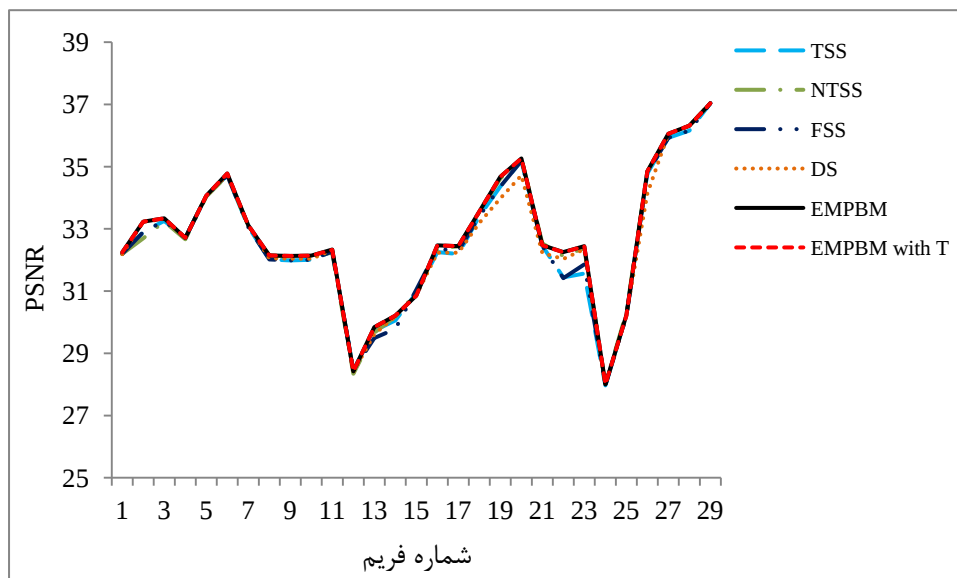


شکل ۵-۷: مقایسه عملکرد بر حسب ΔMSE روی فریم برای فوتبال

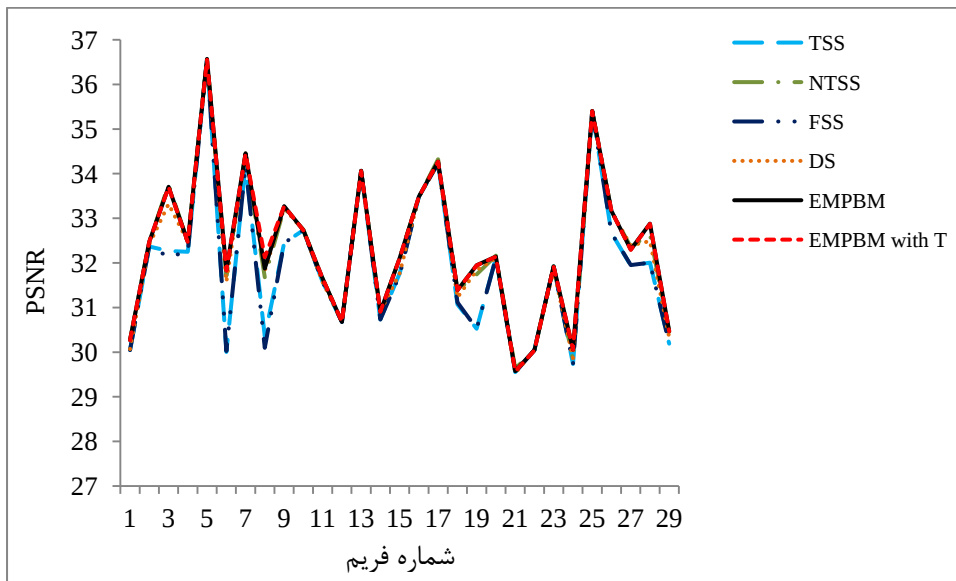


شکل ۵-۸: مقایسه عملکرد بر حسب ΔMSE روی فریم برای اتوبوس

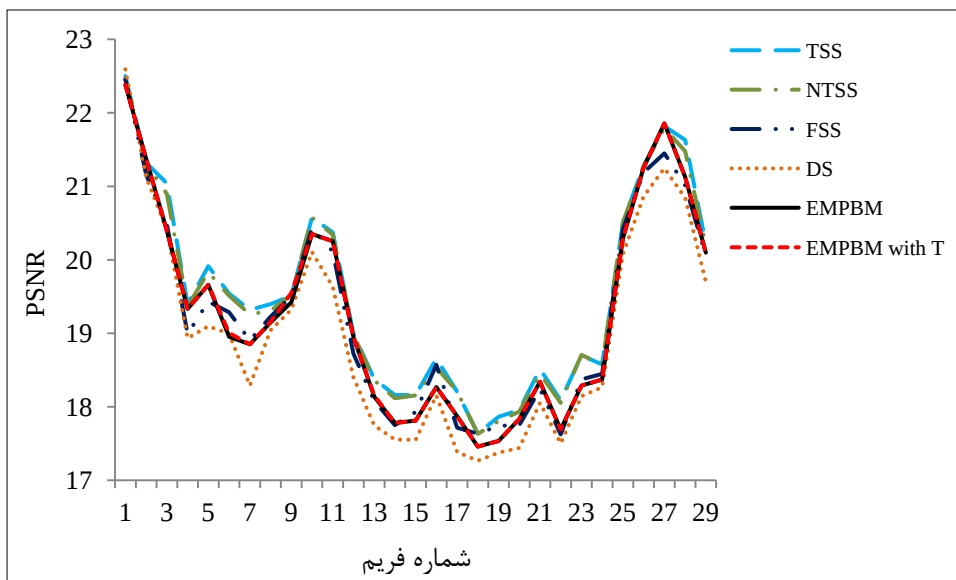
شکل های ۵-۹ تا ۵-۱۶ مقایسه ای از عملکرد الگوریتم پیشنهادی را با الگوریتم های قبلی بر حسب PSNR و $\Delta PSNR$ روی فریم های بازسازی شده برای رشته های آزمون سرکارگر، تلفن ماشین، فوتبال و اتوبوس نشان می دهند.



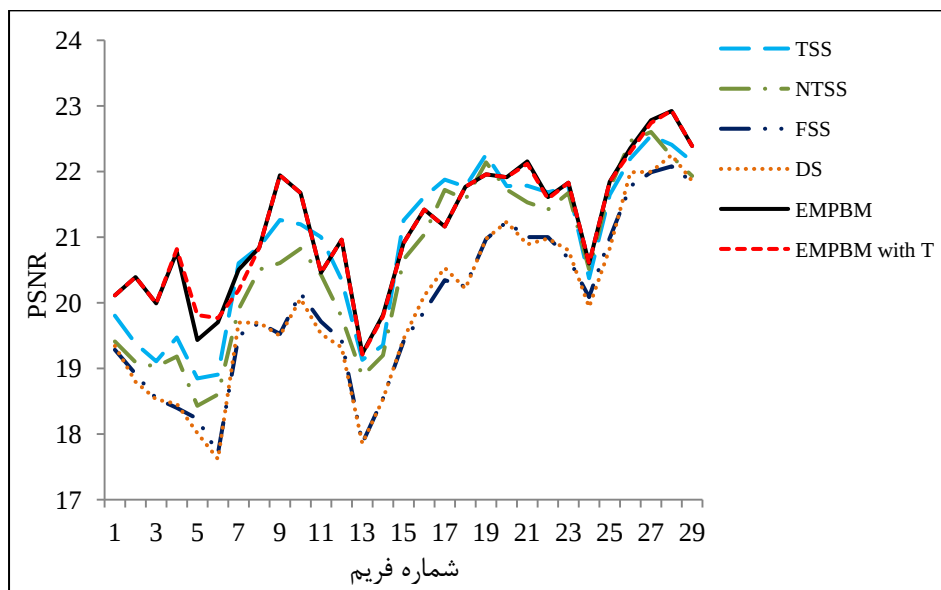
شکل ۵-۹: مقایسه عملکرد بر حسب PSNR روی فریم برای سرکارگر



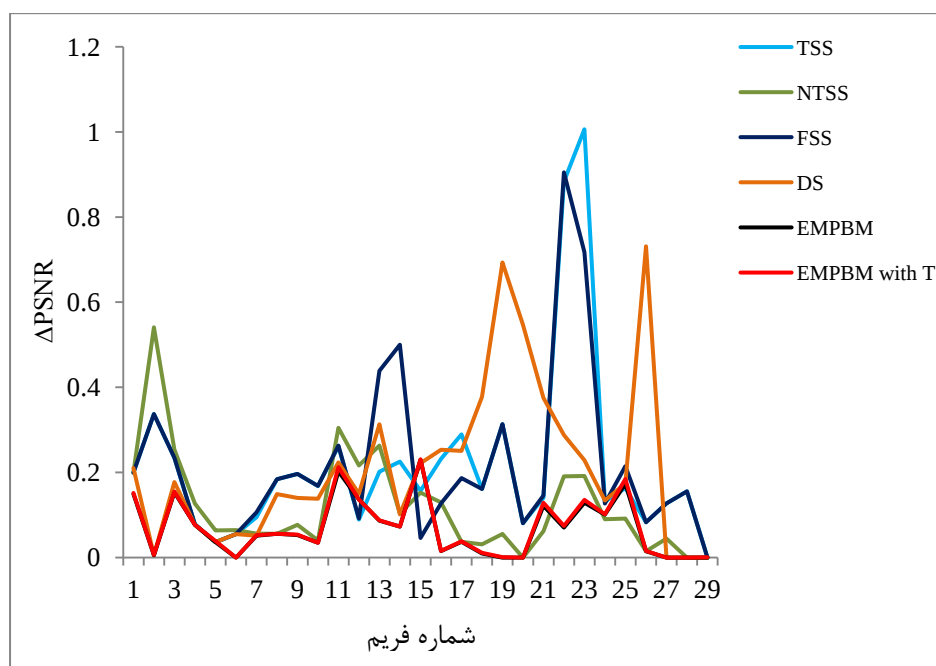
شکل ۵-۱۰: مقایسه عملکرد بر حسب PSNR روی فریم برای تلفن ماشین



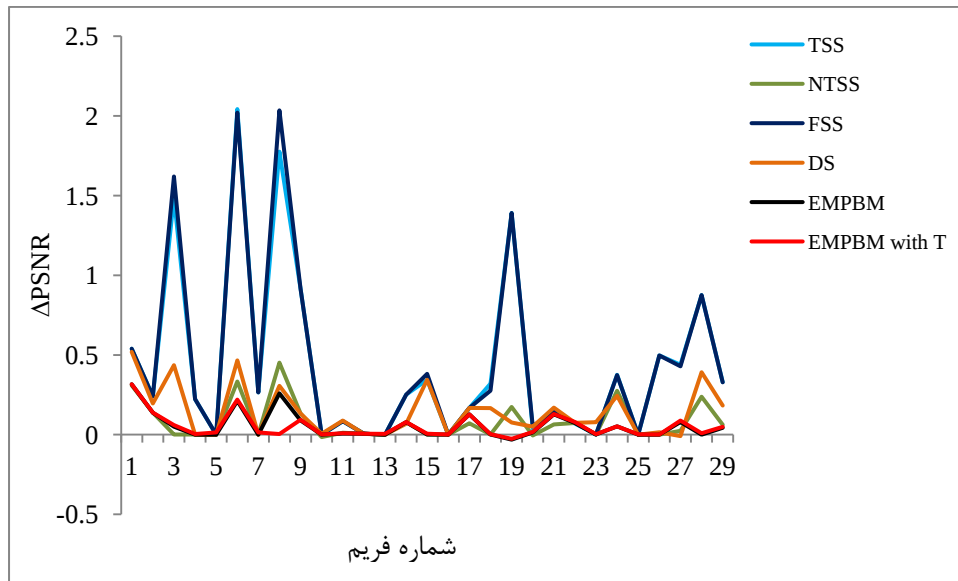
شکل ۵-۱۱: مقایسه عملکرد بر حسب میانگین PSNR برای فوتبال



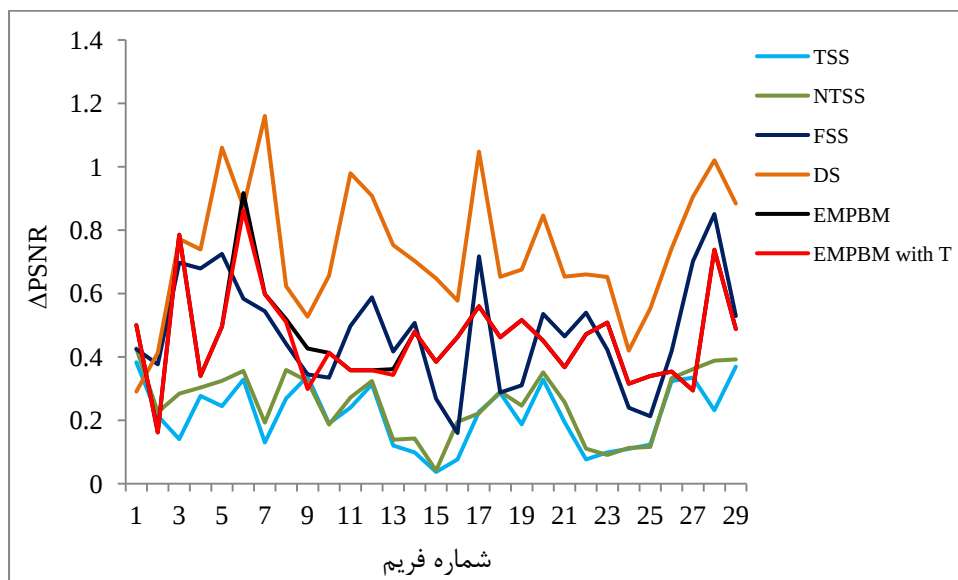
شکل ۵-۱۲: مقایسه عملکرد بر حسب میانگین PSNR برای اتوبوس



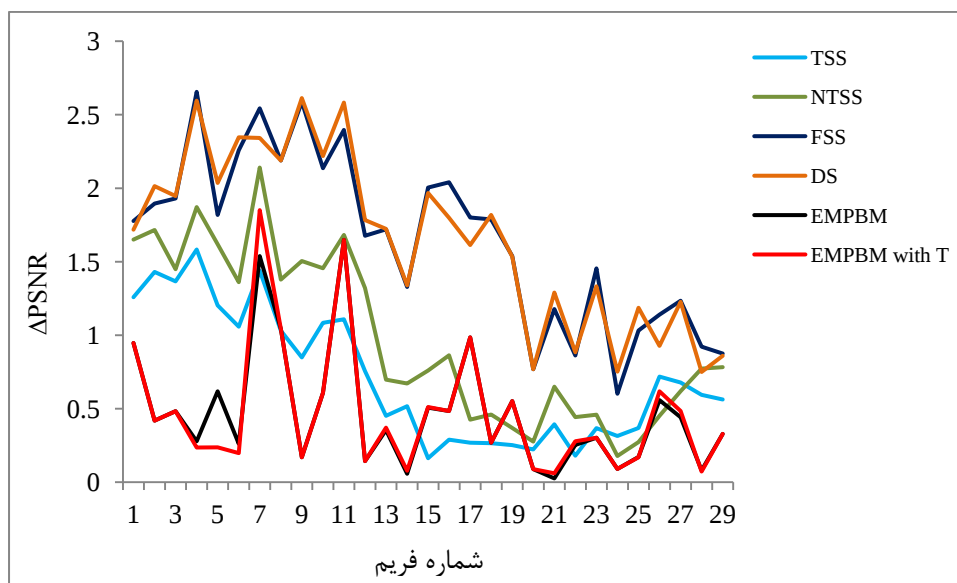
شکل ۵-۱۳: مقایسه عملکرد بر حسب $\Delta PSNR$ روی فریم برای سرکارگر



شکل ۵-۱۴: مقایسه عملکرد بر حسب $\Delta PSNR$ روی فریم برای تلفن ماشین



شکل ۵-۱۵: مقایسه عملکرد بر حسب $\Delta PSNR$ روی فریم برای فوتبال



شکل ۵-۱۶: مقایسه عملکرد بر حسب $\Delta PSNR$ روی فریم برای اتوبوس

۵-۲-۲ نتایج کیفی

کیفیت تصویر بازسازی شده با استفاده از معیارهای عینی^۱ و ذهنی^۲ مورد ارزیابی قرار می‌گیرد. در بخش قبل از معیارهای عینی از قبیل MSE و PSNR برای ارزیابی کیفیت تصویر بازسازی شده استفاده شد. در این بخش از معیار ذهنی استفاده می‌شود. ارزیابی بر اساس معیار ذهنی بصورت مشاهده ویدئوی بازسازی شده توسط گروهی از اشخاص و بیان نظر آن‌ها در مورد کیفیتی که درک کرده‌اند، می‌باشد [۱].

شکل ۵-۱۷ تصویر بازسازی شده فریم شماره ۱۴ رشته آزمون سرکارگر را نشان می‌دهد. بردار حرکت برای این فریم توسط الگوریتم‌های بررسی شده بدست آمده و فریم بر اساس این بردار حرکت بازسازی می‌شود.

^۱Objective
^۲Subjective



(ب) الگوریتم NTSS



(الف) الگوریتم TSS



(د) الگوریتم DS



(ج) الگوریتم FSS



(و) الگوریتم EMPBM با آستانه گذاری



(ه) الگوریتم EMPBM

شکل ۵-۱۷: تصویر بازسازی شده فریم شماره ۱۴ رشته سرکارگر
توسط الگوریتم‌های بررسی شده

به منظور ارزیابی کیفیت تصویر بازسازی شده با استفاده از معیار ذهنی، از ده نفر در
مورد کیفیت تصویر نظرخواهی شد که همگی به تصویر بازسازی شده توسط الگوریتم
EMPBM رأی دادند.

۵-۳ نتایج الگوریتم پیشنهادی دوم

در این بخش نتایج بدست آمده از اعمال الگوریتم پیشنهاد شده را با الگوریتم‌های استاندارد ذکر شده در بخش ۵-۱ مقایسه می‌کنیم. در این مقایسه ۳۰ فریم اول هر رشته آزمون بررسی می‌شوند. برای هر فریم، فریم قبل از آن به عنوان فریم مرجع در نظر گرفته می‌شود. بردار حرکت برای همه فریم‌ها غیر از فریم اول جستجو می‌شود.

۵-۳-۱ نتایج کمی

به منظور ارزیابی عملکرد الگوریتم پیشنهاد شده، مقادیر MSE و PSNR را برای ویدئوهای بازسازی شده محاسبه کرده و نتایج حاصل را با مقادیر بدست آمده از الگوریتم‌های بیان شده مقایسه می‌کنیم. جدول‌های ۵-۵ و ۵-۶ به ترتیب مقادیر میانگین MSE و PSNR را برای ۳۰ فریم اول رشته‌های آزمون نشان می‌دهند. در الگوریتم پیشنهاد شده برای مرحله جستجوی اولیه سه الگوی مختلف ارائه کرده‌ایم و با توجه به ترتیب توضیح، آن‌ها را الگوریتم ۱، الگوریتم ۲ و الگوریتم ۳ می‌نامیم.

جدول ۵-۵: میانگین MSE برای ۳۰ فریم اول رشته‌های آزمون

رشته آزمون	الگوریتم	FS	TSS	NTSS	FSS	DS	الگوریتم ۱ بدون آستانه	الگوریتم ۱ با آستانه	الگوریتم ۲ با آستانه	الگوریتم ۳ با آستانه
Akiyo		۴۰۰۶۵۲	۴۰۰۶۹۱	۴۰۰۶۹۱	۴۰۰۶۹۱	۴۰۰۶۹۱	۴۰۰۶۹۱	۴۰۰۷۱۵	۴۰۰۷۱۵	۴۰۰۷۱۵
Claire		۵۰۱۳۴۱	۵۰۱۵۱۵	۵۰۱۳۴۵	۵۰۱۵۱۳	۵۰۱۳۴۳	۵۰۱۳۴۸	۵۰۱۸۴۷	۵۰۱۸۴۷	۵۰۱۸۴۳
Container		۶۰۶۸۳۶	۶۰۶۸۴۷	۶۰۶۸۶۲	۶۰۶۸۶۴	۶۰۶۸۶۴	۶۰۶۸۷۰	۶۰۶۹۲۷	۶۰۶۹۲۰	۶۰۶۹۲۰
Mother-daughter		۳۰۴۰۷۱	۳۰۴۵۴۵	۳۰۴۰۸۸	۳۰۴۴۰۴	۳۰۴۰۸۰	۳۰۴۰۷۵	۳۰۴۱۶۲	۳۰۴۱۶۲	۳۰۴۱۶۲
Salesman		۱۳۰۸۱۸۷	۱۳۰۸۴۳۳	۱۳۰۸۶۷۴	۱۳۰۸۳۴۰	۱۳۰۸۴۳۳	۱۳۰۸۴۳۳	۱۳۰۸۴۴۸	۱۳۰۸۴۴۸	۱۳۰۸۴۴۸
Foreman		۳۸۰۳۲۲۷	۴۰۰۳۴۸۳	۳۹۰۵۴۱۸	۴۰۰۴۵۳۲	۴۰۰۱۶۱۴	۳۸۰۹۷۲۳	۳۸۰۹۹۱۳	۳۹۰۰۹۸۹	۳۹۰۰۹۴۲
Carphone		۳۹۰۸۲۹۱	۴۴۰۴۰۳۴	۴۰۰۷۳۲۶	۴۴۰۵۵۷۱	۴۱۰۳۲۱۱	۴۰۰۶۰۱۴	۴۰۰۵۵۲۳	۴۰۰۴۹۴۹	۴۰۰۴۹۱۰
Football		۷۲۳۰۷۷۳۱	۷۵۸۰۶۲۱۷	۷۶۴۰۶۹۷۹	۸۰۵۰۴۵۴۳	۸۵۸۰۴۲۶۶	۸۰۲۰۴۰۱۵	۸۰۱۰۶۲۱۱	۸۲۸۰۰۵۹۶	۸۲۸۰۵۳۸۳
Bus		۴۵۸۰۸۷۴۷	۵۴۷۰۴۳۴۱	۵۸۳۰۷۳۹۸	۶۸۲۰۱۶۳۳	۶۸۳۰۹۴۱۶	۶۵۵۰۲۹۱۸	۶۵۴۰۷۳۹۵	۷۳۰۰۷۴۳۰	۷۲۹۰۶۴۸۴

جدول ۵-۶: میانگین PSNR برای ۳۰ فریم اول رشته‌های آزمون

رشته آزمون	الگوریتم	FS	TSS	NTSS	FSS	DS	الگوریتم ۱ بدون آستانه	الگوریتم ۱ با آستانه	الگوریتم ۲ با آستانه	الگوریتم ۳ با آستانه
Akiyo		۴۴.۶۳۴۵	۴۴.۶۳۳۸	۴۴.۶۳۳۸	۴۴.۶۳۳۸	۴۴.۶۳۳۸	۴۴.۶۳۳۸	۴۴.۶۲۸۸	۴۴.۶۲۸۸	۴۴.۶۲۸۸
Claire		۴۱.۸۱۷۲	۴۱.۸۰۸۶	۴۱.۸۱۷۱	۴۱.۸۰۸۲	۴۱.۸۱۵۶	۴۱.۸۱۶۰	۴۱.۷۳۹۳	۴۱.۷۳۹۳	۴۱.۷۳۹۳
Container		۴۳.۲۹۶۲	۴۳.۲۹۶۱	۴۳.۲۹۶۰	۴۳.۲۹۶۰	۴۳.۲۹۶۰	۴۳.۲۹۵۹	۴۳.۲۸۶۶	۴۳.۲۸۶۶	۴۳.۲۸۶۶
Mother-daughter		۴۴.۷۴۵۳	۴۴.۷۲۰۲	۴۴.۷۴۴۸	۴۴.۷۲۷۸	۴۴.۷۴۴۴	۴۴.۷۴۴۷	۴۴.۷۲۴۴	۴۴.۷۲۴۴	۴۴.۷۲۴۴
Salesman		۳۸.۳۸۲۰	۳۸.۳۷۸۴	۳۸.۳۷۵۴	۳۸.۳۸۰۱	۳۸.۳۷۸۴	۳۸.۳۷۸۴	۳۸.۳۷۷۲	۳۸.۳۷۷۲	۳۸.۳۷۷۲
Foreman		۳۲.۸۱۴۶	۳۲.۵۹۷۹	۳۲.۶۹۵۴	۳۲.۵۹۸۱	۳۲.۶۰۴۲	۳۲.۷۵۶۴	۳۲.۷۵۴۴	۳۲.۷۴۴۷	۳۲.۷۴۴۳
Carphone		۳۲.۴۰۵۷	۳۱.۹۶۳۴	۳۲.۳۲۱۹	۳۱.۹۵۰۷	۳۲.۲۶۱۷	۳۲.۳۳۹۲	۳۲.۳۴۳۸	۳۲.۳۴۸۵	۳۲.۳۴۸۸
Football		۱۹.۷۴۱۷	۱۹.۵۲۴۹	۱۹.۴۸۷۸	۱۹.۲۶۵۲	۱۹.۰۰۳۹	۱۹.۲۸۹۰	۱۹.۲۹۳۴	۱۹.۱۴۲۷	۱۹.۱۴۴۴
Bus		۲۱.۶۲۴۸	۲۰.۹۰۸۶	۲۰.۶۴۹۲	۱۹.۹۶۴۵	۱۹.۹۶۴۲	۲۰.۱۴۳۰	۲۰.۱۴۹۸	۱۹.۷۰۳۸	۱۹.۶۵۵۸

جدول‌های ۵-۷ و ۵-۸ به ترتیب زمان پردازش و سرعت الگوریتم را برای ۳۰ فریم اول

۵ رشته آزمون نشان می‌دهند.

جدول ۵-۷: زمان مورد نیاز برای پردازش ۳۰ فریم اول رشته‌های آزمون بر حسب ثانیه

رشته آزمون	الگوریتم	FS	TSS	NTSS	FSS	DS	الگوریتم ۱ بدون آستانه	الگوریتم ۱ با آستانه	الگوریتم ۲ با آستانه	الگوریتم ۳ با آستانه
Akiyo		۴.۷۷۱۹	۲.۵۷۸۹	۱.۳۳۶۰	۱.۵۳۷۱	۱.۳۴۲۸	۱.۵۸۵۸	۰.۴۵۴۳	۰.۴۰۵۷	۰.۳۹۷۲
Claire		۴.۸۵۵۰	۲.۶۳۵۵	۱.۵۲۲۸	۱.۶۵۱۵	۱.۴۸۴۹	۱.۵۹۱۶	۰.۴۷۶۱	۰.۴۴۲۴	۰.۴۳۱۳
Container		۴.۹۱۷۴	۲.۷۵۶۵	۱.۵۱۲۵	۱.۶۷۴۱	۱.۴۹۰۴	۱.۶۵۶۱	۰.۶۱۹۴	۰.۵۵۸۷	۰.۵۴۴۹
Mother-daughter		۴.۷۶۴۱	۲.۶۱۱۸	۱.۴۴۴۰	۱.۵۹۰۰	۱.۴۲۱۰	۱.۵۵۲۸	۰.۵۲۰۱	۰.۴۸۲۰	۰.۴۵۴۷
Salesman		۴.۹۴۶۸	۲.۷۳۲۱	۱.۴۴۱۹	۱.۶۳۴۷	۱.۴۲۳۱	۱.۵۹۷۰	۰.۷۱۱۶	۰.۶۹۹۹	۰.۶۷۷۸
Foreman		۴.۸۱۷۳	۲.۶۳۹۵	۱.۷۸۳۲	۱.۸۱۰۰	۱.۶۸۴۲	۱.۶۸۹۹	۱.۴۷۸۰	۱.۶۲۶۵	۱.۵۷۲۶
Carphone		۴.۹۰۲۷	۲.۶۶۹۴	۱.۸۱۹۳	۱.۸۴۴۵	۱.۷۲۰۵	۱.۷۶۷۳	۱.۴۳۳۷	۱.۵۰۴۵	۱.۳۹۹۶
Football		۴.۹۰۱۰	۲.۶۷۰۴	۲.۴۵۶۷	۲.۵۴۰۴	۲.۵۸۰۱	۲.۳۹۸۹	۲.۳۴۸۹	۳.۱۴۷۲	۳.۱۱۷۷
Bus		۴.۹۳۶۴	۲.۶۷۸۰	۲.۵۳۶۰	۲.۴۲۰۱	۲.۴۶۲۰	۲.۰۲۳۸	۲.۰۶۵۷	۲.۵۵۰۷	۲.۴۹۸۷

جدول ۵-۸: سرعت پردازش

الگوریتم رشته آزمون	FS	TSS	NTSS	FSS	DS	الگوریتم ۱ بدون آستانه	الگوریتم ۱ با آستانه	الگوریتم ۲ با آستانه	الگوریتم ۳ با آستانه
Akiyo	۱	۱.۸۵۰۴	۳.۵۷۱۸	۳.۱۰۴۵	۳.۵۵۳۷	۳.۰۰۹۱	۱۰.۵۰۳۹	۱۱.۷۶۲۱	۱۲.۰۱۳۸
Claire	۱	۱.۸۴۲۲	۳.۱۸۸۲	۲.۹۳۹۸	۳.۲۶۹۶	۳.۰۵۰۴	۱۰.۱۹۷۴	۱۰.۹۷۴۲	۱۱.۲۵۶۷
Container	۱	۱.۷۸۳۹	۳.۲۵۱۲	۲.۹۳۷۳	۳.۲۹۹۴	۲.۹۶۹۳	۷.۹۳۹۰	۸.۸۰۱۵	۹.۰۲۴۴
Mother-daughter	۱	۱.۸۲۴۱	۳.۲۹۹۲	۲.۹۹۶۳	۳.۳۵۲۶	۳.۰۶۸۱	۹.۱۶۰۰	۹.۸۸۴۰	۱۰.۴۷۷۵
Salesman	۱	۱.۸۱۰۶	۳.۴۳۰۸	۳.۰۲۶۱	۳.۴۷۶۱	۳.۰۹۷۶	۶.۹۵۱۷	۷.۰۶۷۹	۷.۲۹۸۳
Foreman	۱	۱.۸۲۵۱	۲.۷۰۱۵	۲.۶۶۱۵	۲.۸۶۰۳	۲.۸۵۰۶	۳.۲۵۹۳	۲.۹۶۱۸	۳.۰۶۳۳
Carphone	۱	۱.۸۳۶۶	۲.۶۹۴۸	۲.۶۵۸۰	۲.۸۴۹۶	۲.۷۷۴۱	۳.۴۱۹۶	۳.۲۵۸۷	۳.۵۰۲۹
Football	۱	۱.۸۳۵۳	۱.۹۹۵۰	۱.۹۲۹۲	۱.۸۹۹۵	۲.۰۴۳۰	۲.۰۸۶۵	۱.۵۵۷۳	۱.۵۷۲۰
Bus	۱	۱.۸۴۳۳	۱.۹۴۶۵	۲.۰۳۹۸	۲.۰۰۵۰	۲.۴۳۹۲	۲.۳۸۹۷	۱.۹۳۵۳	۱.۹۷۵۶

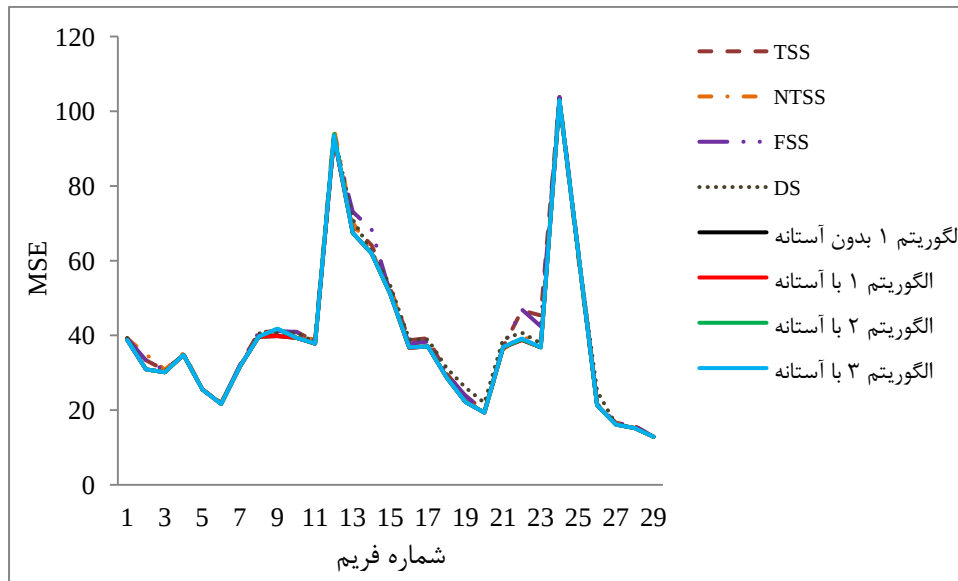
با توجه به جداول ۵-۵ و ۵-۶ مشاهده می‌شود که مقدار PSNR و MSE بدست آمده از الگوریتم پیشنهادی برای رشته‌های آزمون متعلق به کلاس‌های ۱ و ۲ تقریباً مشابه است، و همانطور که انتظار داشتیم با اعمال روش ZMP نتایج کمی افت کرد. رشته‌های آزمون متعلق به کلاس ۳ شامل رشته‌های با جابجایی زیاد و رشته‌های با جابجایی متوسط می‌باشند. همه الگوریتم‌های پیشنهادی در این بخش برای رشته‌های آزمون با جابجایی متوسط نتیجه بهتری داشته‌اند؛ اما در رشته‌های آزمون با جابجایی بسیار زیاد، به دلیل اینکه بردار حرکت اکثر ماکروبلوک‌ها با یکدیگر متفاوت است و بردارهای حرکت بسیاری در جهت‌های مختلف وجود دارد، تنها الگوریتم ۱ به دلیل جستجوی کاملتری که در مرحله جستجوی اولیه انجام می‌دهد، نتیجه بهتری را دارا می‌باشد. بررسی موقعیت بردار حرکت

پیشگویی در این قبیل رشته‌های آزمون موقعیت کمینه سراسری را بدرستی مشخص نمی‌کند و به همین علت نتیجه در الگوریتم‌های ۲ و ۳ کمی افت می‌کند. اعمال روش ZMP نیز به دلیل تعداد کم ماکروبلوک‌های ثابت در این رشته‌های آزمون تأثیر قابل توجهی روی نتایج نمی‌گذارد.

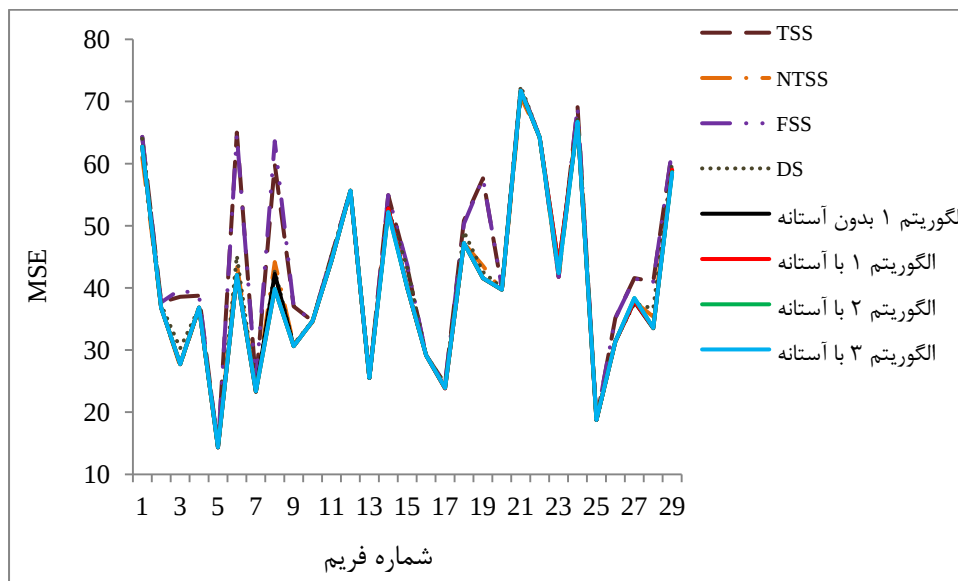
با توجه به جداول ۵-۷ و ۵-۸ مشاهده می‌شود که زمان جستجوی بلوک منطبق در الگوریتم‌های پیشنهادی برای رشته‌های آزمون کلاس‌های ۱ و ۲ به دلیل استفاده از روش ZMP به میزان زیاد و برای رشته‌های آزمون کلاس ۳ به میزان کمتری کاهش یافته است. این کاهش زمان نشان دهنده کاهش تعداد نقاط جستجو و کاهش پیچیدگی می‌باشد. در الگوریتم‌های ۲ و ۳ به دلیل اینکه جستجو بر اساس موقعیت بردار حرکت ماکروبلوک چپ انجام می‌گیرد و در رشته‌های آزمون با جابجایی بسیار زیاد، بردار حرکت اکثر ماکروبلوک‌ها با یکدیگر متفاوت است، موقعیت کمینه سراسری در مرحله جستجوی اولیه به درستی مشخص نشده و تعداد نقاط بیشتری در مرحله جستجوی محلی بررسی می‌شوند و از این رو زمان جستجوی بلوک منطبق افزایش می‌یابد.

با در نظر گرفتن همه نتایج بدست آمده، ملاحظه می‌شود که می‌توان مصالحه‌ای بین کیفیت ویدئوی بازسازی شده و زمان جستجوی بلوک منطبق انجام داد. از آنجایی که در مخابرات ویدئویی تلفن همراه زمان پردازش مهم است، با توجه به PSNR و زمان پردازش بدست آمده، الگوریتم‌های پیشنهادی می‌توانند کارآیی مناسبی داشته باشند.

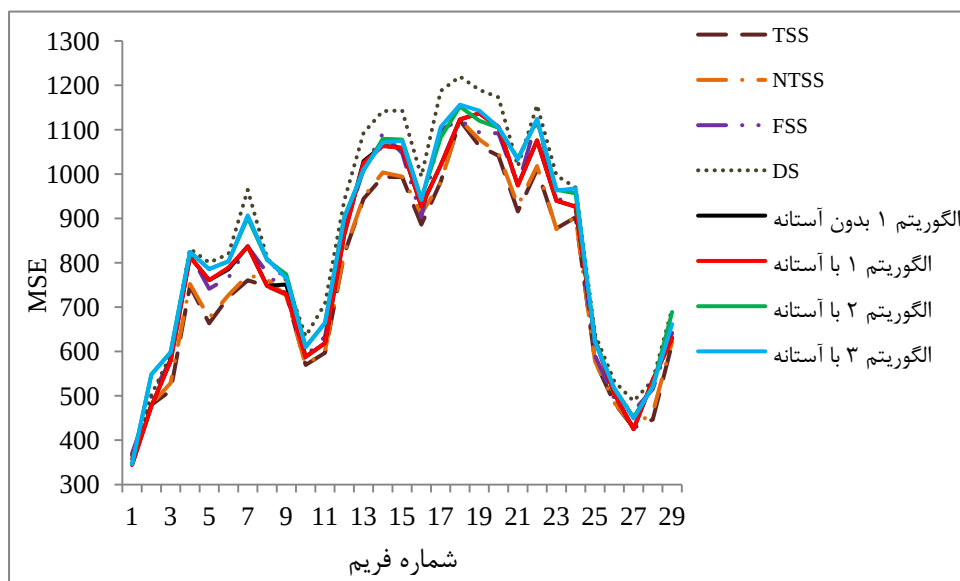
شکل‌های ۵-۱۸ تا ۵-۲۱ مقایسه‌ای از عملکرد الگوریتم‌های بررسی شده را بر اساس میانگین مجذور خطا (MSE) روی فریم‌های بازسازی شده برای رشته‌های آزمون سرکارگر، تلفن ماشین، فوتبال و اتوبوس نشان می‌دهد.



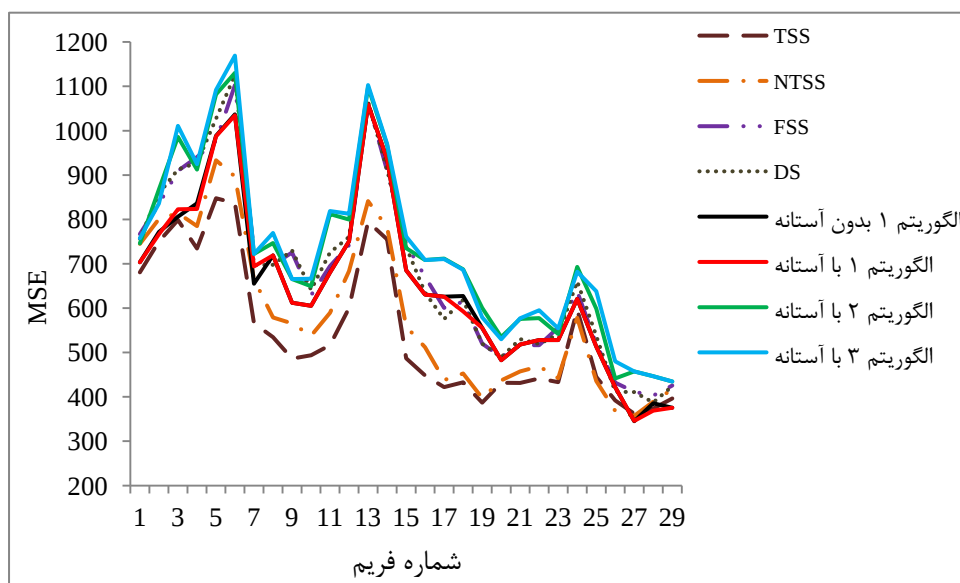
شکل ۵-۱۸: مقایسه عملکرد بر حسب MSE روی فریم برای سرکارگر



شکل ۵-۱۹: مقایسه عملکرد بر حسب MSE روی فریم برای تلفن ماشین



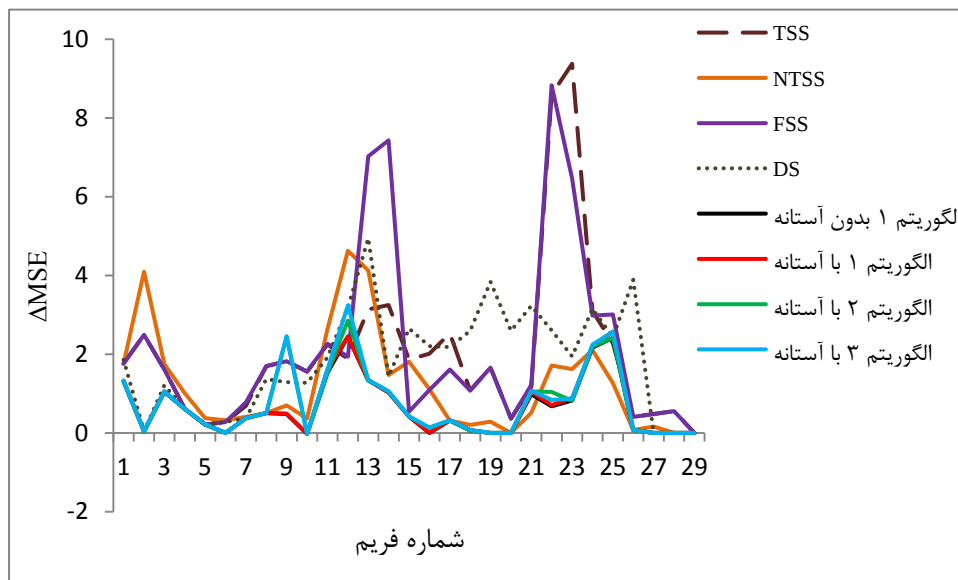
شکل ۵-۲۰: مقایسه عملکرد بر حسب MSE روی فریم برای فوتبال



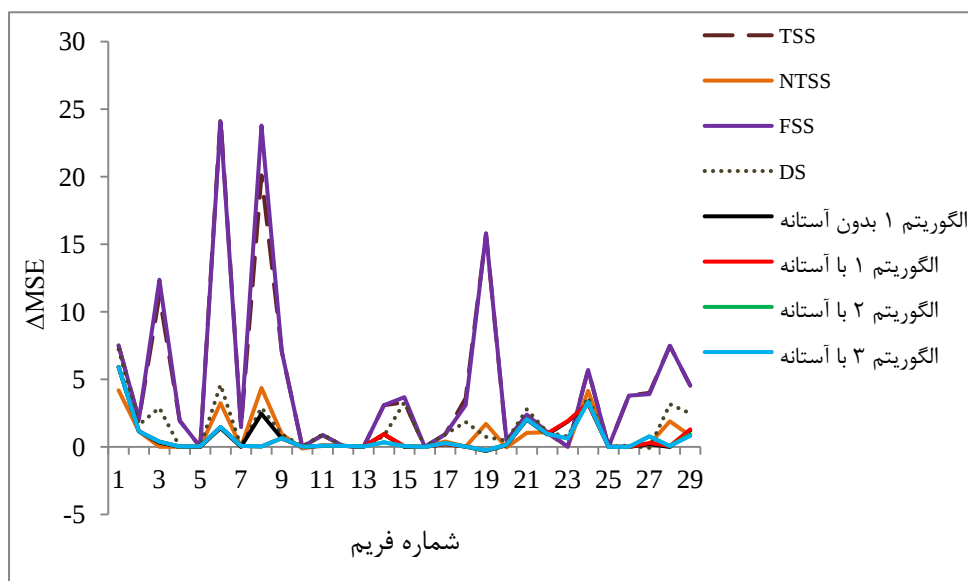
شکل ۵-۲۱: مقایسه عملکرد بر حسب MSE روی فریم برای اتوبوس

به منظور درک بهتر عملکرد الگوریتم پیشنهادی اختلاف مقادیر MSE همه الگوریتم‌های با مقادیر MSE الگوریتم FS محاسبه شده و نمودار این اختلافات رسم می‌شود.

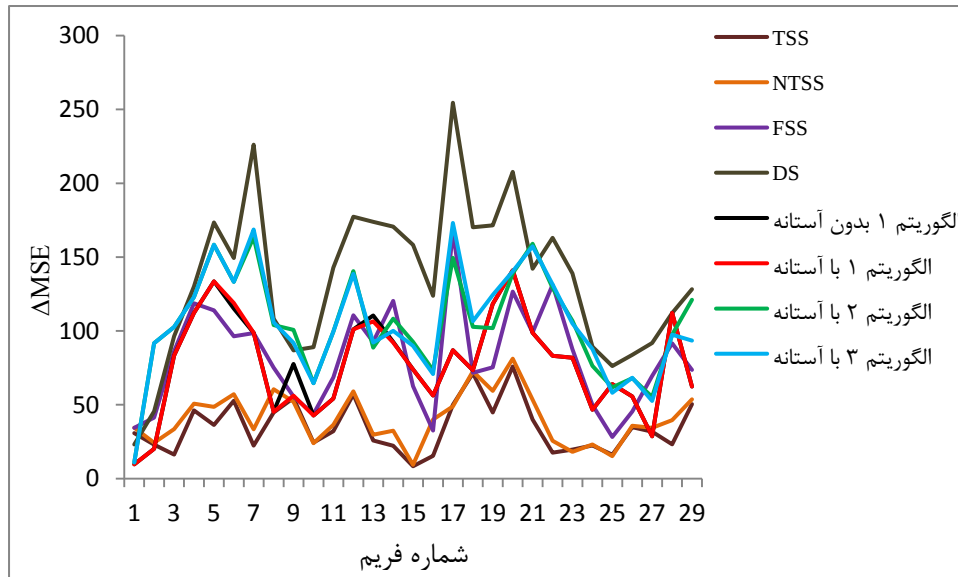
شکل‌های ۵-۲۲ تا ۵-۲۵ نمودار ΔMSE همه الگوریتم‌ها برای رشته‌های آزمون سرکارگر، تلفن ماشین، فوتبال و اتوبوس را نشان می‌دهند.



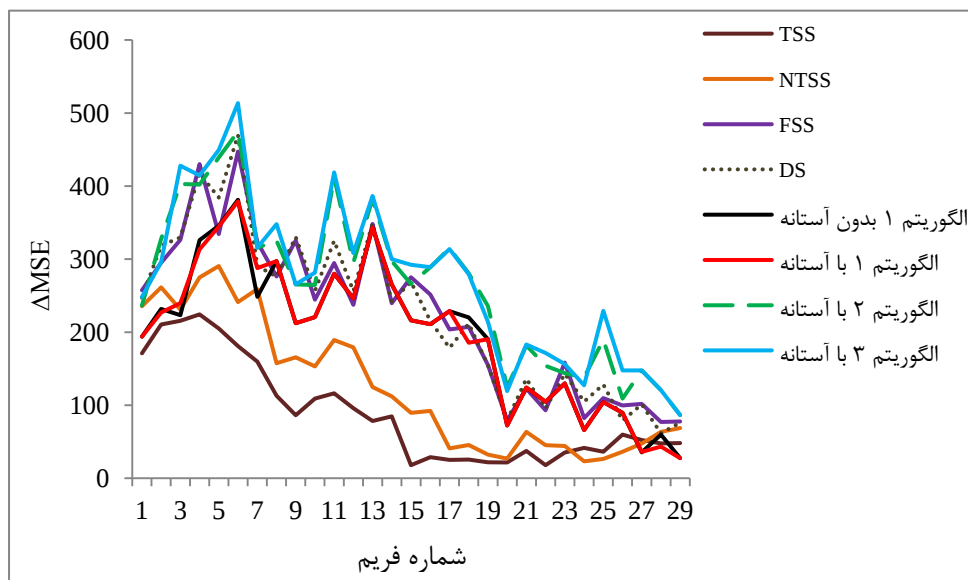
شکل ۵-۲۲: مقایسه عملکرد بر حسب ΔMSE روی فریم برای سرکارگر



شکل ۵-۲۳: مقایسه عملکرد بر حسب ΔMSE روی فریم برای تلفن ماشین

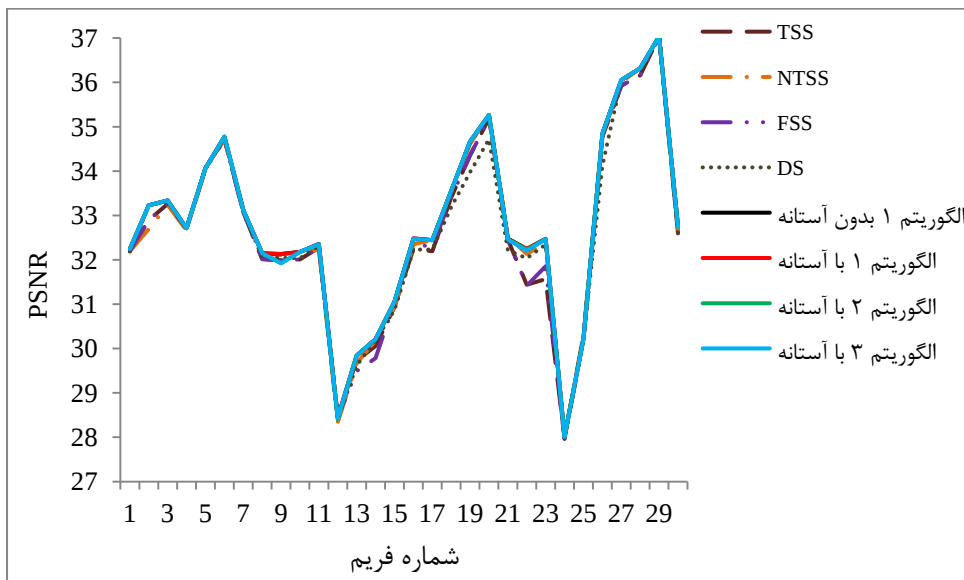


شکل ۵-۲۴: مقایسه عملکرد بر حسب ΔMSE روی فریم برای فوتبال

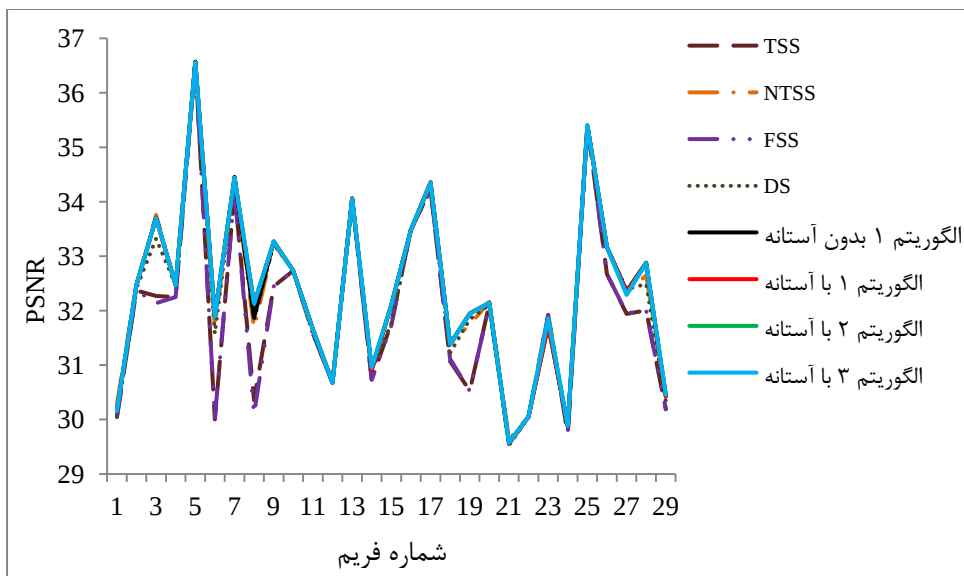


شکل ۵-۲۵: مقایسه عملکرد بر حسب ΔMSE روی فریم برای اتوبوس

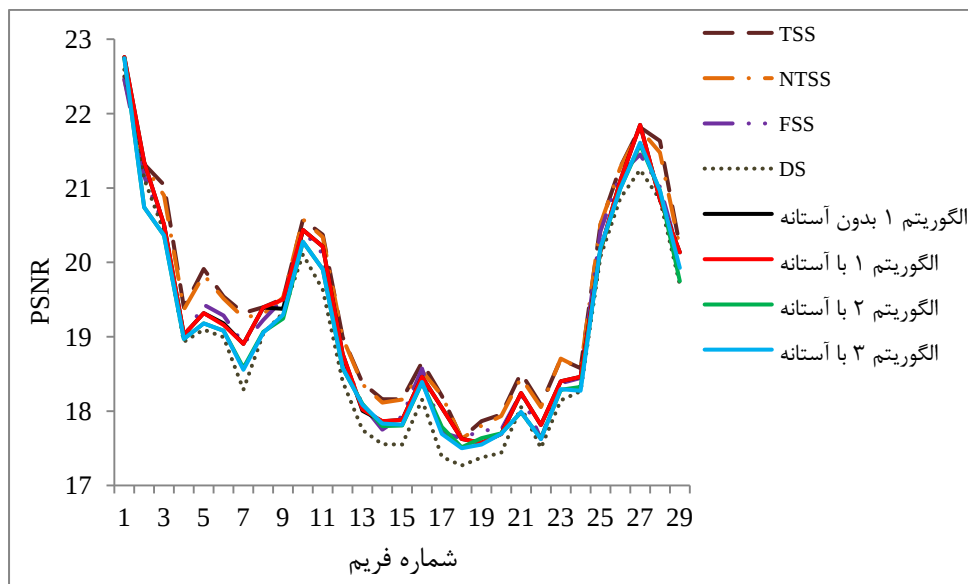
شکل‌های ۵-۲۶ تا ۵-۳۳ مقایسه‌ای از عملکرد الگوریتم پیشنهادی را با الگوریتم‌های قبلی بر حسب PSNR و Δ PSNR روی فریم‌های بازسازی شده برای رشته‌های آزمون سرکارگر، تلفن ماشین، فوتبال و اتوبوس نشان می‌دهند.



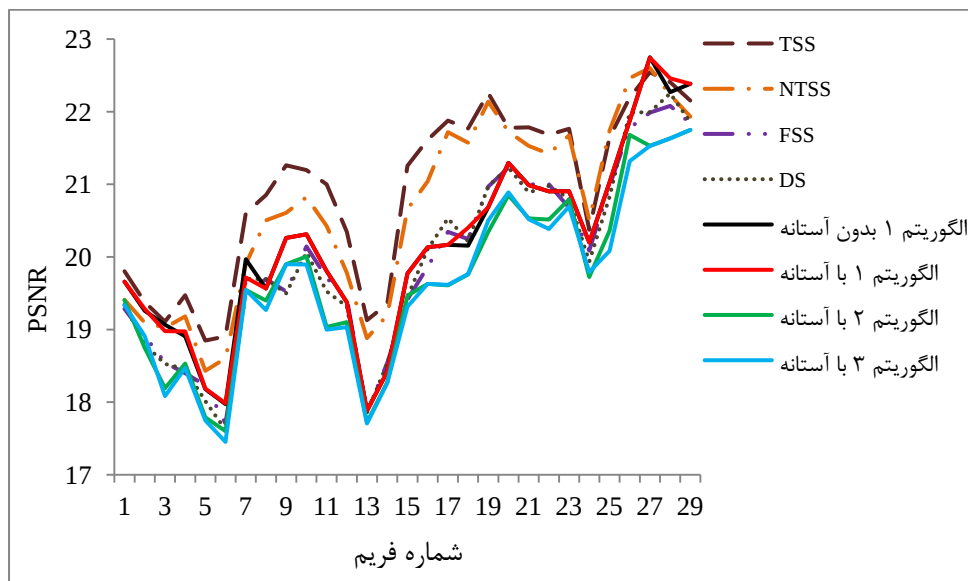
شکل ۵-۲۶: مقایسه عملکرد بر حسب میانگین PSNR برای سرکارگر



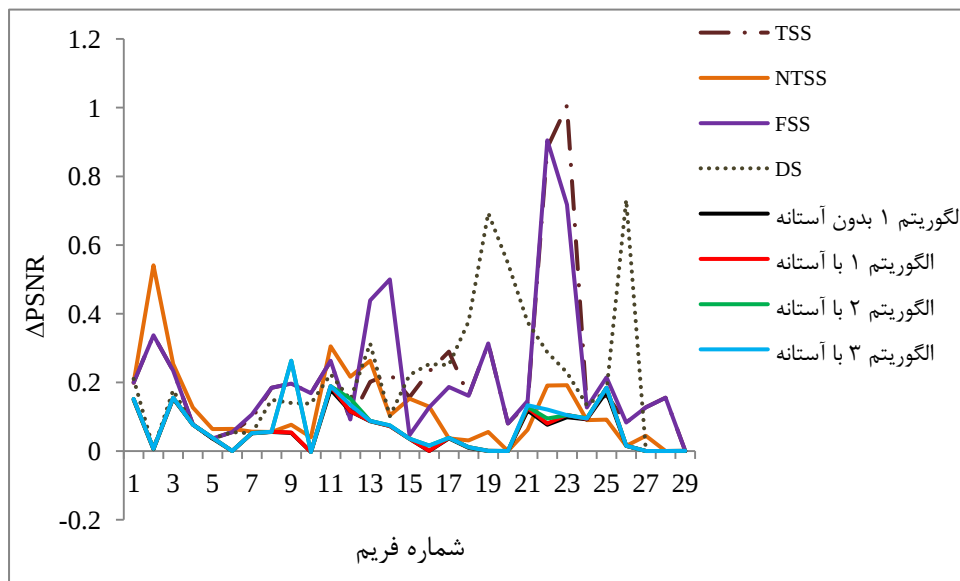
شکل ۵-۲۷: مقایسه عملکرد بر حسب میانگین PSNR برای تلفن ماشین



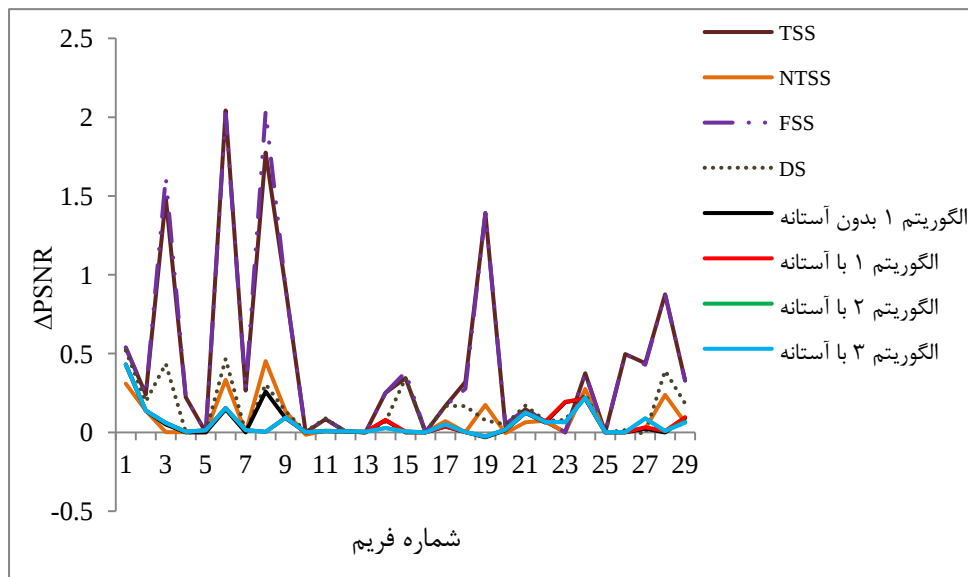
شکل ۵-۲۸: مقایسه عملکرد بر حسب میانگین PSNR برای فوتبال



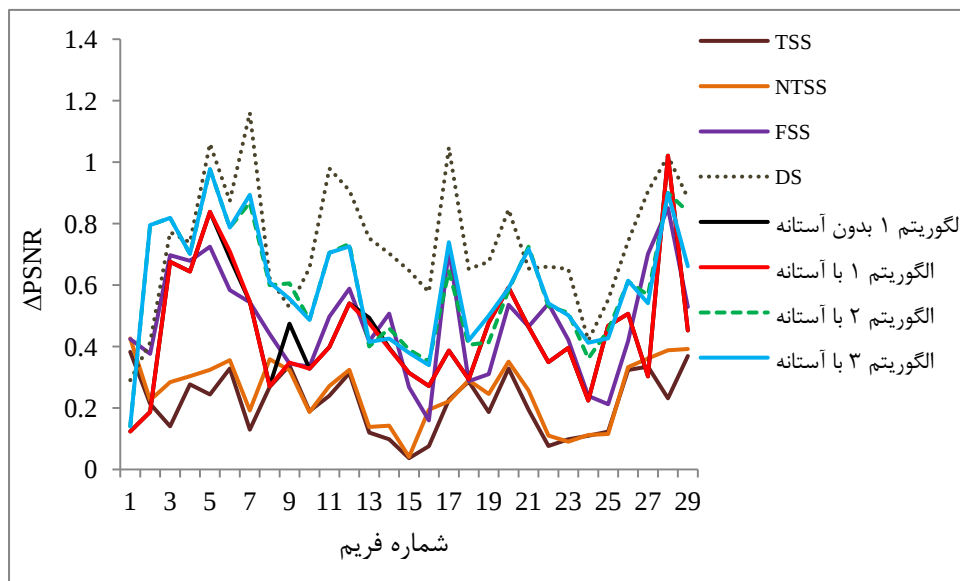
شکل ۵-۲۹: مقایسه عملکرد بر حسب میانگین PSNR برای اتوبوس



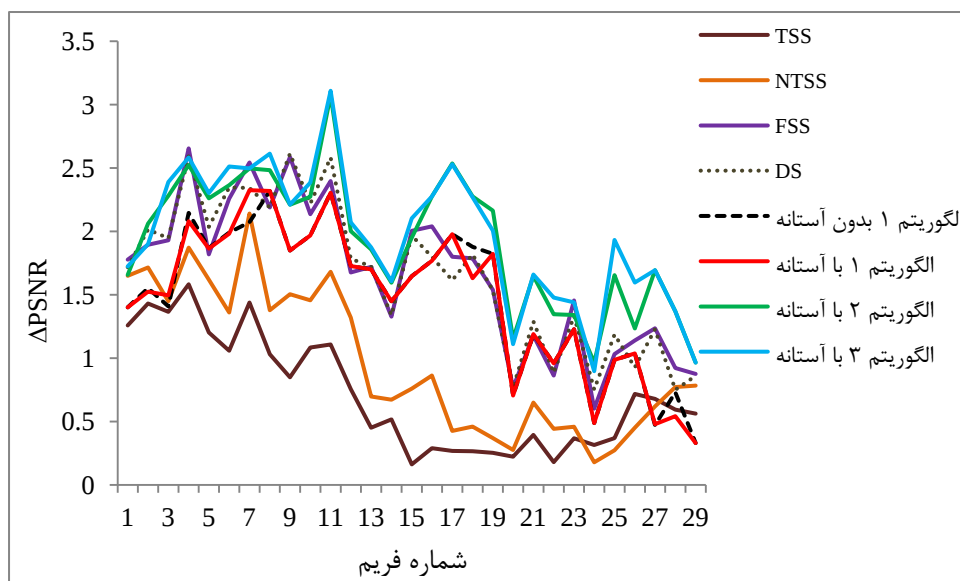
شکل ۵-۳۰: مقایسه عملکرد بر حسب $\Delta PSNR$ روی فریم برای سرکارگر



شکل ۵-۳۱: مقایسه عملکرد بر حسب $\Delta PSNR$ روی فریم برای تلفن ماشین



شکل ۵-۳۲: مقایسه عملکرد بر حسب $\Delta PSNR$ روی فریم برای فوتبال



شکل ۵-۳۳: مقایسه عملکرد بر حسب $\Delta PSNR$ روی فریم برای اتوبوس

۵-۳-۲ نتایج کیفی

در این بخش، با استفاده از معیار ذهنی کیفیت ویدئوهای بازسازی شده توسط الگوریتم‌های پیشنهادی را بررسی می‌کنیم. شکل ۵-۳۴ تصویر بازسازی شده فریم شماره ۱۹ رشته آزمون تلفن ماشین را نشان می‌دهد. بردار حرکت برای این فریم توسط الگوریتم‌های بررسی شده بدست آمده و فریم بر اساس این بردار حرکت بازسازی می‌شود.



(ب) الگوریتم NTSS



(الف) الگوریتم TSS



(د) الگوریتم DS



(ج) الگوریتم FSS



(و) الگوریتم ۱ با آستانه

(ه) الگوریتم ۱ بدون آستانه



(ح) الگوریتم ۳ با آستانه

(ز) الگوریتم ۲ با آستانه

شکل ۵-۳۴: تصویر بازسازی شده فریم شماره ۱۹ رشته تلفن ماشین
توسط الگوریتم‌های بررسی شده

به منظور ارزیابی کیفیت تصویر بازسازی شده با استفاده از معیار ذهنی، از ده نفر در مورد کیفیت تصویر نظرخواهی شد که همه این اشخاص رأی به مشابه بودن کیفیت تصویر با تعدادی از الگوریتم‌ها و بهتر بودن نسبت به تعدادی دیگر دادند.

فصل ششم

نتیج گیری و پیشنهادات

۶ فصل ششم

۶-۱ مقدمه

در این فصل ابتدا نتیجه‌گیری در مورد کارهای انجام شده در این تحقیق بیان می‌شود و سپس در مورد پیشنهادات برای کارهای آتی بحث می‌گردد.

۶-۲ نتیجه‌گیری

این تحقیق با هدف ارائه الگوریتم تطبیق بلوک جدیدی برای تخمین حرکت انجام گرفت که تعداد نقاط جستجو را کاهش داده در حالیکه کیفیت ویدئوی بازسازی شده مطلوبی را نتیجه می‌دهد.

کاربرد این الگوریتم در واحد تخمین حرکت سیستم‌های رمزگذاری ویدئو است. از آنجایی که بیشترین محاسبات در این سیستم‌ها مربوط به واحد تخمین حرکت است، با ارائه الگوریتم‌های سریع می‌توان پیچیدگی محاسباتی و در نتیجه زمان محاسبات را کاهش داد و به این ترتیب، سیستم رمزگذاری ویدئو در کاربردهای بلادرنگ مانند مخابرات ویدئویی تلفن همراه قابل استفاده خواهد بود.

در این تحقیق دو الگوریتم برای تطبیق بلوک پیشنهاد شد. در الگوریتم اول از الگوی صلیبی و افقی و موقعیت دو بردار حرکت همسایه بالا و چپ در مرحله جستجوی اولیه و از الگوی مربعی 3×3 در مرحله جستجوی محلی استفاده کردیم و نشان دادیم که زمان محاسبات به میزان قابل توجهی کاهش یافت که بیانگر کاهش تعداد نقاط جستجو است، در حالیکه PSNR نزدیک و گاهی بهتر از الگوریتم‌های تطبیق بلوک استاندارد را نتیجه می‌دهد.

در الگوریتم دوم، سه الگوی مختلف در مرحله جستجوی اولیه پیشنهاد شد. در الگوی اول، از الگوی مربعی بزرگ استفاده کردیم و اندازه گام‌های افقی و عمودی را بر اساس مؤلفه‌های افقی و عمودی بردار حرکت همسایه چپ تعیین نمودیم. در الگوی دوم، ابتدا چهار نقطه از الگوی مربعی بزرگ که در محدوده بردار حرکت همسایه چپ قرار داشتند، را بررسی کرده و سپس با توجه به قانون بیان شده الگوی جستجوی نقاط را کامل کردیم. در الگوی سوم، تنها چهار نقطه در محدوده بردار حرکت همسایه چپ را بررسی کردیم. در مرحله جستجوی محلی از الگوی مربعی 3×3 به منظور پیدا کردن نقطه MME استفاده نمودیم. نتایج بدست آمده در این سه الگو تقریباً مشابه است. در آن‌ها زمان جستجو برای اکثر ویدئوهای آزمون کاهش یافت در حالیکه PSNR مشابه و گاهی بهتر از الگوریتم‌های تطبیق بلوک استاندارد را نتیجه داد.

بطور کلی با بررسی نتایج بدست آمده از دو الگوریتم پیشنهادی مشاهده می‌کنیم که این الگوریتم‌ها برای رشته‌های آزمون کلاس ۳ بخوبی جواب می‌دهند و برای کلاس‌های دیگر اگرچه PSNR کمی کاهش می‌یابد، اما به دلیل سرعت بالای الگوریتم می‌توان از آن چشم‌پوشی کرده و بین سرعت و PSNR مصالحه کرد.

۳-۶ پیشنهادات

- در الگوریتم‌های پیشنهاد شده از اندازه بلوک 16×16 استفاده شده است. هر چه اندازه بلوک کوچکتر باشد، دقت جستجو بیشتر می‌شود اما زمان جستجو افزایش می‌یابد. به منظور بهبود PSNR پیشنهاد می‌شود که از اندازه بلوک متغیر استفاده شود. به این صورت که اگر مقدار خطای تطبیق برای ماکروبلوک با اندازه 16×16 از مقدار آستانه از پیش تعیین شده بیشتر باشد اما بردار حرکت

(۰,۰) باشد، ماکروبلوک به چهار بلوک 8×8 تقسیم شده و بردار حرکت برای آن‌ها بدست می‌آید. آستانه به همان نسبت کوچک می‌شود؛ اگر مقدار خطای هر بلوک از آستانه کمتر باشد و باز هم بردار حرکت (۰,۰) باشد، بلوک به چهار بلوک کوچکتر تقسیم می‌شود. این روند ادامه می‌یابد تا زمانی که بردار حرکت بدرستی تعیین شود. با این روش جابجایی‌های کوچک نیز شناسایی می‌شود و PSNR افزایش می‌یابد، اما زمان جستجو کمی بیشتر می‌شود.

- در اکثر رشته‌های ویدئویی در مخابرات تلفن همراه جابجایی کوچکی وجود دارد و بیشتر ماکروبلوک‌ها متعلق به پس‌زمینه می‌باشند. به منظور پرهیز از بررسی این ماکروبلوک‌ها پیشنهاد می‌شود که شیء‌های متحرک را از پس‌زمینه جدا کرده و بردار حرکت را توسط الگوریتم‌های پیشنهادی برای ماکروبلوک‌های متعلق به شیء‌های متحرک بدست آورد. در این روش باید از دو فریم مرجع (فریم قبل و بعد از فریم فعلی) استفاده شود.

٧ مراجع

- [1] Al-Mualla M. E, Canagarajah C. N. and Bull D. R., (2002), PhD. Thesis, "Video coding for mobile communications", university of Bristol U.K.
- [2] Bull D. R., Canagarajah C. N., and Nix A. R., (1999), "Insights into Mobile Multimedia Communications", Signal Processing and Its Applications. Academic Press, London.
- [3] Hoang D. T., Long P. M., and Vitter J. S., (1998), "Efficient cost measures for motion estimation at low bit rates", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 8, No. 4, pp. 488–500.
- [4] Limb J. O., and Murphy J. A., (1975), "Measuring the speed of moving objects from television signals", IEEE Transactions on Communication, Vol. 23, No. 4, pp.474–478.
- [5] Cafforio C., and Rocca F., (1976), "Methods for measuring small displacements of television images", IEEE Transactions on Information Theory, Vol. 22, No. 5, pp. 573–579.
- [6] Yamaguchi H., (1989), "Iterative method of movement estimation for television signals", IEEE Transactions on Communications, Vol. 37, No. 12, pp. 1350 –1358.
- [7] Ming H. M., Chou Y. M., and Cheng S. C., (1997), "Motion estimation for video coding standards", Journal of VLSI Signal Processing: Systems for Signal, Image, and Video Technology, pp. 113–136.
- [8] Netravali N., and Robbins J. D., (1979), "Motion compensated television coding: Part I", Bell System Technical Journal, Vol. 58, No. 3, pp. 631– 670.
- [9] Walker D. R., and Rao K. R., (1984), "Improved pel-recursive motion compensation", IEEE Transactions on Communications, Vol. 32, No. 10, pp. 1128–1134.
- [10] Dufaux F., and Moscheni F., (1995), "Motion estimation techniques for digital TV: A review and a new contribution", Proceedings of the IEEE, Vol. 83, No. 6, pp. 858–875.
- [11] Kuglin C. D., and Hines D. C., (1975), "The phase correlation image alignment method", In Proceedings of the IEEE International Conference on Cybernetics and Society, Vol. 45, No. 13, pp. 163–165.
- [12] Thomas G. A., (1987), "Television motion measurement for DATV and other applications", Technical Report 1987/11, British Broadcasting Corporation (BBC) Research Department.
- [13] Snell & Wilcox Ltd, Alchemist Ph.C D: a phase correlation 10-bit motion compensated standards converter for digital I/O, <http://www.snellwilcox.com/productguide/linker1/alc2index.html>.
- [14] Jain J. R., and Jain A. K., (1981), "Displacement measurement and its application in interframe image coding". IEEE Transactions on Communications, Vol. 29, No. 12, pp. 1799–1808.
- [15] Ericsson S., (1985), "Fixed and adaptive predictors for hybrid predictive/transform coding", IEEE Transactions on Communications, Vol. 33, No. 12, pp. 1291–1302.

- [16] Girod B., (1993), "Motion-compensating prediction with fractional-pel accuracy", IEEE Transactions on Communications, Vol. 41, No. 4, pp. 604–612.
- [17] Manikandan M., Vijayakumar P., and Ramadass N., (2006), "Motion Estimation Method for Video Compression–An Overview", IFIP International Conference on Wireless and Optical Communications Networks, Vol. 29, No. 12, pp. 1799-1808.
- [18] Chan E., (1993), "Review of block matching based motion estimation algorithms for video compression", IEEE Conference on Electrical and Computer Engineering, vol. 1, pp. 151-154.
- [19] Pandian S. I. A., Bala Dr. G. J., and George B. A., (2011), "A study on block matching algorithm for motion estimation", International Journal on Computer Science and Engineering (IJCSE), vol. 3, pp. 34-44.
- [20] Koga T., Iinuma K., Hirano A., Iijima Y., and Ishiguro T., (1981), "Motion-compensated interframe coding for video conferencing", in Proc. Nat. Telcommun. Conf., pp. 5.3.1~5.3.5, New Orleans, LA.
- [21] Lu J., and Liou M. L., (1997), "A Simple and Efficient Search Algorithm for Block-Matching Motion Estimation", IEEE Trans. Circuits and Systems for Video Technology, Vol. 7, No. 2, pp. 429-433.
- [22] Kim J. N., and Choi T. S., (1998), "A fast three step search algorithm with minimum checking points using unimodal error surface assumption", IEEE Transactions on consumer electronics, Vol. 44, No. 3, pp. 638-648.
- [23] Li R., Zeng B., and Liou M. L., (1994), "A New Three-Step Search Algorithm for Block Motion Estimation", IEEE Trans. Circuits and Systems for Video Technology, Vol. 4, No. 4, pp. 438-442.
- [24] Rao K. R., and Hwang J. J., (1996) "Techniques and Standards for Image, Video and Audio Coding", Englewood Cliffs, NJ: Prentice Hall.
- [25] Chau L. P., and Jing X., (2003), "Efficient three step search algorithm for block motion estimation in video coding", IEEE International conference on acoustics, speech, and signal processing (ICASSP), Vol. 3, pp. III-421-4.
- [26] Jing X., and Chau L. P., (2004), "An efficient three-step search algorithm for block motion estimation", IEEE Transactions on multimedia, Vol. 6, No. 3, pp. 435-438.
- [27] Po L. M., and Ma W. C., (1996), "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation", IEEE Trans. Circuits and Systems For Video Technology, Vol 6, No. 3, pp. 313-317.
- [28] Liu K., Qiu Q., and Zhang Z., (2011), "A novel fast motion estimation algorithm based on block-matching", IEEE Cross strait quad-regional radio science and wireless technology conference, Vol. 2, pp. 1402-1405.
- [29] Zhu S., and Ma K. K., (2000), "A new diamond search algorithm for fast block-matching motion estimation", IEEE Transactions on image processing, Vol. 9, No. 2, pp. 287-290.
- [30] Babu D. V., Subramanian P., and Karthikeyan C., (2006), "Performance analysis of block matching algorithms for highly scalable video compression", IEEE International symposium on Ad Hoc and Ubiquitous Computing, pp. 179-182.

- [31] Nie Y., and Ma K. K., (2002), "Adaptive rood pattern search for fast block-matching motion estimation", IEEE Transactions on image processing, Vol. 11, No. 12, pp. 1442-1449.
- [32] Zhu S., and Ma K. K., (1998), "A new star search algorithm for fast blockmatching motion estimation," in Proc. Workshop on Very Low Bitrate Coding (VLBV), pp. 173-176.
- [33] Zhao H., Yu X. B., Sun J. H., Sun C., and Cong H. Z., (2008), "An enhanced adaptive rood pattern search algorithm for fast block-matching motion estimation", IEEE Congress on image and signal processing, Vol.1, pp. 416-420.
- [34] Ahmed Z., Hussain A. J., and Al-Jumeily D., (2011), "Mean predictive block matching (MPBM) for block-matching motion estimation", IEEE 3rd European workshop on visual information processing (EUVIP), pp. 67-72.
- [35] Available at <http://media.xiph.org/video/derf/>

Abstract

Block matching motion estimation is a key component in video compression. Motion estimation (ME) is a core part in video coding, which can effectively remove temporal redundancies among the adjacent video frames to achieve high coding efficiency.

In this thesis, we propose two block matching algorithms to reduce the matching block searching time and improve the quality of the reconstructed video. The algorithms consist of two stages: the initial search and the local search. In the first algorithm, an adaptive rood pattern with unequal arms and position of left and above neighboring motion vectors are used in the initial search; and the 3×3 square pattern is used in the local search. In the second algorithm, we have proposed three different patterns in the initial search. First, a large square pattern, in which the horizontal and vertical step sizes are determined according to horizontal and vertical components of the left neighboring motion vector. Second, the pattern of four points of the large square pattern, which are located in the left neighboring motion vector range; and then complete the search pattern based on error values obtained from these points. Third, the pattern of four points of the large square pattern without considering of matching error value of these points, which are located in the left neighboring motion vector range. The 3×3 square pattern is used in the local search.

The results of the proposed algorithms for 9 test video sequences and comparison with standard block matching algorithms shows that the block matching searching time is reduced and algorithms are about 2~12 times faster than that of full search algorithm and the PSNR, especially for video sequence with large motion is improved.

Key words: motion estimation, video coding, block matching, adaptive rood pattern, square pattern.



Shahrood University of Technology

Faculty of Electrical and Robatic

**A new video compression method for mobile communication using motion
estimation**

Mahsa Esmailpour Quchani

Supervisor:

Dr. Omidreza Marouzi

Date: Bahman 91