



فشرده سازی تصاویر ویدئویی بر مبنای الگوریتم ژنتیک

محمد رضا سروش مهر^۱ و محمد علی منتظری^۲

^۱ دانشگاه آزاد اسلامی، واحد خمینی شهر، soroushmehr@iaukhsh.ac.ir

^۲ دانشکده برق و کامپیوتر، دانشگاه صنعتی اصفهان، montazeri@cc.iut.ac.ir

چکیده - در این مقاله بر مبنای الگوریتم ژنتیک روشی جدید برای یافتن بهترین بلوک در بخش تخمین حرکت کد کننده های ویدئویی پیشنهاد می گردد. در روش پیشنهادی، جمعیت اولیه بر اساس همبستگی مکانی و زمانی بردارهای حرکت انتخاب می گردد. همچنین از آنجا که درصد زیادی از بلوکهای تصویر ایستان هستند یا حرکت آنها به ناحیه کوچکی در اطراف ناحیه جستجو محدود می شود جمعیت اولیه شامل این بلوکها نیز می باشد. برای کاهش بیشتر حجم محاسبات، در هر یک از مراحل الگوریتم مقدار تابع برازش (fitness) با آستانه ای که از قبل تعریف می شود مقایسه می گردد تا در صورت بیشتر بودن مقدار تابع از مقدار آستانه، جستجو متوقف گردد. نتایج اعمال الگوریتم پیشنهادی بر روی تصاویر استاندارد نشان می دهد که این الگوریتم در مقایسه با الگوریتمهای مشابه حجم کمتری از محاسبات را نیاز دارد در حالیکه PSNR آن مشابه الگوریتمهای متداول در این زمینه یا بهتر از آنها می باشد.

کلید واژه - الگوریتم ژنتیک، تخمین حرکت، فشرده سازی ویدئویی.

کیفیت تصاویر بازسازی شده، الگوریتمهای دسته اول استفاده بیشتری دارند. الگوریتمهای دسته اول را می توان شامل روش جستجوی بازگشتی پیکسل [۳]، روشهایی که از تبدیل affine استفاده می کنند [۴] و روش انطباق بلوک در نظر گرفت. روشهای مبتنی بر انطباق بلوک نیز تنوع زیادی دارند که در بخش دوم به مرور تعدادی از این الگوریتمها می پردازیم.

الگوریتم پیشنهادی شده در این مقاله از دسته الگوریتمهای انطباق بلوکی حرکت می باشد که بر مبنای الگوریتم ژنتیک طراحی شده است. در انتخاب جمعیت اولیه از خاصیت ایستان و شبه ایستان بودن حرکت بلوکها [۵] و همچنین از خاصیت همبستگی زمانی و مکانی بردارهای حرکت استفاده شده است. همچنین از آستانه گذاری روی تابع fitness برای کاهش حجم محاسبات نیز کمک گرفته شده است.

در ادامه و در بخش دوم به مرور روش انطباق بلوکی حرکت و تعدادی از روشهای مطرح در این زمینه می پردازیم. در بخش سوم مراحل الگوریتم پیشنهادی معرفی می گردد. نتایج اعمال الگوریتم روی تصاویر استاندارد با نتایج الگوریتمهای موجود در بخش چهارم مقایسه می شود. بخش آخر نیز به نتیجه گیری اختصاص دارد.

۱- مقدمه

تخمین حرکت یکی از بخشهای مهم کد کننده های استاندارد نظیر MPEG1/2/4 و H261/263/264 می باشد. وظیفه این بخش کاهش همبستگی بین فریمهای متوالی یک تصویر ویدئویی به منظور فشرده سازی این تصاویر است. به طور مثال در ویدئو-کنفرانسها یا سیستمهای جدید تلفنی که قابلیت انتقال ویدئو را نیز دارند نیاز است که حجم تصاویر کم شده تا انتقال آن سریعتر انجام شود. در عین حال باید کیفیت تصاویر تا حد قابل قبولی حفظ شود. از آنجا که حجم زیادی از محاسبات کد کننده های ویدئویی به بخش تخمین حرکت اختصاص می یابد تحقیقات بسیار زیادی راجع به این بخش انجام شده و روشهای بسیار متنوعی تا کنون در این زمینه ارائه گردیده است. هدف کلی این تحقیقات ارائه الگوریتمهای سریعی بوده که در آنها کیفیت تصویر بازسازی شده مناسب باشد. این روشها را می توان به دو دسته کلی تقسیم کرد. دسته اول شامل روشهایی است که تخمین را در حوزه زمان انجام می دهند و دسته دوم روشهایی را شامل می شود که تخمین را در حوزه فرکانس انجام داده [۱] و یا از حوزه فرکانس برای بهبود تخمین استفاده می کنند [۲] بخاطر پیچیدگی محاسباتی کم و قابل قبول بودن

۲- تخمین حرکت بر مبنای انطباق بلوک

در روش تخمین حرکت بر مبنای انطباق بلوک، فریم جاری به بلوکهای مربعی غیر همپوشان تقسیم می‌گردد. سپس بازای هر بلوک، ناحیه ای به نام ناحیه جستجو در فریم مرجع تعیین می‌گردد. این ناحیه به اندازه w پیکسل از هر یک از ۴ لبه بلوک نظیر بلوک جاری در فریم مرجع گسترده می‌شود. بلوکها نیز $N \times N$ ، در نظر گرفته می‌شود. از بین بلوکهای این ناحیه، بلوکی که بیشترین انطباق را با بلوک جاری دارد بعنوان بلوک منطبق شناخته می‌شود. برای تعیین بلوک منطبق، یک تابع معیار در نظر گرفته می‌شود. معمولاً تابع معیار جمع قدر مطلق خطا (SAE) بخاطر کارایی و پیاده سازی ساده، استفاده بیشتری دارد. برداری که بین دو نقطه نظیر بلوک جاری و بلوک منطبق رسم می‌شود بردار حرکت نامگذاری می‌گردد. بردار حرکت را می‌توان طبق رابطه ۱ بدست آورد.

$$MV = \operatorname{argmin}_{(x,y)} \sum_{m=1}^N \sum_{n=1}^N \left| I_{cur}(x_0 + m, y_0 + n) - I_{ref}(x_0 + x + m, y_0 + y + n) \right| \quad (1)$$

where $-w \leq x, y \leq w$

در رابطه ۱، $I_{ref}(i, j)$ ، $I_{cur}(i, j)$ به ترتیب بیانگر میزان روشنایی پیکسل واقع در مکان (i, j) از فریم جاری و فریم مرجع می‌باشد. در ضمن فرض شده که مختصات پیکسل واقع در گوشه چپ و بالای بلوک مورد نظر (x_0, y_0) است. MV نیز مشخص کننده میزان جابجایی بلوک جاری نسبت به بلوک نظیر آن در فریم مرجع می‌باشد. یک روش ساده برای پیدا کردن بردار حرکت این است که کل ناحیه جستجو، مورد بررسی قرار گیرد. این روش را روش جستجوی کامل (FS) می‌نامند. با روش FS، بلوکی که دارای کمترین خطا (می‌نیمم سراسری) باشد پیدا می‌شود. این الگوریتم درعین سادگی، نیازمند حجم محاسبات بسیار زیادی می‌باشد و بکار بردن آن، انجام بلادرنگ تخمین را مشکل می‌نماید. به این دلیل تا کنون الگوریتمهای زیادی برای افزایش سرعت جستجو و انجام بلادرنگ بخش تخمین حرکت ارائه شده است. در برخی از الگوریتمها الگوی ثابتی برای یافتن نقطه بهینه بکار برده می‌شود. از جمله این الگوریتمها می‌توان به الگوریتم جستجوی سه مرحله ای (3SS) [6]، الگوریتم جستجوی چهار مرحله ای (4SS) [5]، الگوریتم جستجو با الگوی لوزی شکل (DS) [7] و الگوریتم جستجو با الگوی هشت وجهی [8] اشاره نمود. هر چند سرعت این الگوریتمها نسبت به الگوریتم FS بسیار بیشتر است ولی در این الگوریتمها

ممکن است بجای می‌نیمم سراسری، می‌نیمم محلی پیدا شود. از این رو در پاره ای از الگوریتمها فرض می‌شود که در ناحیه کوچکی اطراف می‌نیمم سراسری با دورشدن از نقطه می‌نیمم، میزان خطا افزایش می‌یابد [9]. بنابراین در برخی از الگوریتمها سعی می‌شود که در ابتدا بردار حرکت بلوک پیشگویی شده، در اطراف مکان پیشگویی شده جستجو انجام گیرد. اساس این پیشگوییها همبستگی زمانی و مکانی بردارهای حرکت است. در مرجع [10]، با در نظر گرفتن همبستگی مکانی و زمانی بردارهای حرکت، ناحیه جستجو بطور وقتی برای هر بلوک تعیین می‌گردد تا جستجوی کمتری نسبت به FS انجام شود. در الگوریتم FBME [11] از منطق فازی برای پیشگویی بردار حرکت استفاده شده است. در الگوریتم مرجع شماره [12] و بهبود یافته آن یعنی الگوریتم FAME [13] از همبستگی زمانی و مکانی بردارهای حرکت برای پیشگویی بردار حرکت یک بلوک استفاده شده است. در LGSA [14] و 4GS [15] از الگوریتم ژنتیک برای جستجو استفاده شده است.

در این مقاله با در نظر گرفتن همبستگی مکانی و زمانی بردارهای حرکت، با استفاد از الگوریتم ژنتیک، روشی سریع برای یافتن بردارهای حرکت پیشنهاد می‌شود.

۳- استفاده از الگوریتم ژنتیک در تخمین حرکت

در این بخش ابتدا مروری بر الگوریتم 4GS خواهیم داشت و پس از آن مراحل الگوریتم پیشنهادی ارائه می‌گردد.

۳-۱- الگوریتم 4GS

در الگوریتم 4GS ابتدا جمعیت اولیه بطور تصادفی انتخاب می‌شود. جمعیت اولیه شامل ۱۶ بلوک است که در اطراف مرکز ناحیه جستجو متمرکز می‌باشند. در این الگوریتم، هر یک از بردارهای حرکت، معرف یک کروموزوم است. از آنجا که w برابر با ۷ انتخاب شده است برای نمایش هر کروموزوم به ۸ بیت نیاز می‌باشد (چهار بیت برای کد کردن راستای افقی بردار و ۴ بیت برای راستای عمودی آن). در این الگوریتم از تابع میانگین قدرمطلق خطا (MAE) به عنوان تابع fitness استفاده شده است. مراحل الگوریتم 4GS پس از انتخاب جمعیت اولیه به شرح زیر است:

- ۱- تولید مثل (reproduction): در این مرحله بر اساس چرخ رولت (roulette wheel) از هر نسل کروموزومهایی برای تولید مثل انتخاب می‌شوند. یعنی ابتدا رابطه (۲) محاسبه می‌شود.

مرجع بسیار نزدیک است. این خاصیت تحت عنوان همبستگی مکانی و زمانی بردارهای حرکت معروف می باشد. در شکل ۱، بخشی از بلوکهای یک فریم نمایش داده شده است

✓	✓	✓	✓
✓	B1	B2	✓
✓	B3	?	✓

شکل ۱: بخشی از بلوکهای یک فریم

با توجه به اینکه تعیین بردارهای حرکت بلوکها، از سمت چپ و بالای یک فریم شروع و ردیف به ردیف انجام می شود، هنگام تعیین بردار حرکت بلوکی مانند BC، بردار حرکت بلوکهایی که با علامت ✓ مشخص شده اند معلوم می باشد. در این مقاله، بردار حرکت بلوکها را به عنوان کروموزوم در نظر گرفته، برای تعیین بردار حرکت بلوک BC، بردارهای حرکت بلوکهای B1, B2, B3 و همچنین بردار حرکت بلوک نظیر BC در فریم مرجع و بردار (0, 0) را به عنوان جمعیت اولیه در نظر می گیریم. این بردارها را با MV_i که $0 \leq i \leq 4$ است نمایش می دهیم. در این حالت MV_0, MV_1 به ترتیب نظیر بردار (0, 0) و بردار حرکت بلوک نظیر BC در فریم مرجع است.

تابع fitness را طبق رابطه ۶ تعریف می کنیم.

$$f(C) = MAX - SAE(C) \quad (6)$$

در رابطه ۶، C معرف کروموزوم و MAX حداکثر مقداری است که تابع جمع قدرمطلق خطا (SAE) می تواند داشته باشد. بطور مثال در صورتیکه بلوکها را 16×16 و هر پیکسل را ۸ بیتی فرض کنیم SAE، ۱۶ بیتی و حداکثر آن 65535 خواهد شد. کم کردن یک عدد ۱۶ بیتی از عدد 65535 نظیر متمم کردن بیتهای آن عدد است. در ضمن در این رابطه بر خلاف رابطه ۳ نیازی به مرتب کردن تابع fitness کروموزومها نیست و بلوکهایی که مقدار SAE آنها کمتر باشد تابع fitness آنها بزرگتر خواهد شد و طبق رابطه ۲، بازه بزرگتری به آنها اختصاص می یابد و شانس انتخابشان برای نسل بعدی افزایش می یابد در حالیکه طبق روابط ۲ و ۳ که در الگوریتم 4GS از آنها استفاده می شود، کروموزومهایی که MAE آنها بزرگتر باشد احتمال انتخابشان در نسل بعدی بیشتر است.

در الگوریتم PVGSA نیز از چرخ رولت برای انتخاب کروموزوم استفاده می گردد. با توجه به مطالب گفته شده، در ادامه، مراحل الگوریتم PVGSA ذکر شده است.

۱- انتخاب بردارهای $(0 \leq i \leq 4)$ MV_i به عنوان

$$r_i = \left(\sum_{k=0}^{i-1} f_k, \sum_{k=0}^i f_k \right) \quad (2)$$

در رابطه ۲، f_i تابع fitness نظیر کروموزوم i ام می باشد. طبق این رابطه به هر کدام از کروموزومها متناسب با مقدار تابع fitness بازه ای اختصاص می یابد. تابع fitness طبق رابطه زیر محاسبه می گردد.

$$f_i(C_i) = \begin{cases} MAE(C_i) & MAE(C_i) < MAE(C_n) \\ MAE(C_n) & MAE(C_i) \geq MAE(C_n) \end{cases} \quad (3)$$

در رابطه ۳، C_i ها معرف کروموزومها هستند و C_n کروموزومی است که تابع fitness آن، n امین کوچکترین مقدار را دارد.

۲- جهش (mutation): در این مرحله مقدار هر کروموزوم طبق رابطه زیر تغییر می کند.

$$X'_n = X_n + x_{mut} \times size \quad (4)$$

$$Y'_n = Y_n + y_{mut} \times size$$

(X'_n, Y'_n) مقدار جدید کروموزوم و (X_n, Y_n) مقدار قبلی آن می باشد. در این رابطه مقدار size برای سه نسل اول برابر با ۲ و برای نسل چهارم ۱ می باشد. x_{mut} و y_{mut} نیز طبق روابط زیر محاسبه می گردد.

$$x_{mut} = (-1)^i \lceil j/4 \rceil, y_{mut} = (-1)^k \lceil l/4 \rceil \quad (5)$$

$$i = (p+3) \text{ div } 4, j = (p+3) \text{ mod } 4$$

$$k = p \text{ div } 4, l = p \text{ mod } 4, p = n \text{ mod } 8$$

در روابط بالا، متغیر n اندیس کروموزوم است.

۳- کروموزومهای جهش یافته و کروموزومهای قبل از جهش بر اساس تابع MAE مرتب می شوند. نیمی از کروموزومها که تابع fitness آنها کمتر است برای نسل بعدی انتخاب می گردند.

۴- عملیات بالا چهار بار تکرار می شود.

۳-۲- الگوریتم پیشنهادی

در این قسمت الگوریتم پیشنهادی که آنرا "الگوریتم جستجوی ژنتیکی بر مبنای بردارهای پیشگو (PVGSA)" نامگذاری کرده ایم معرفی می شود.

قبل از توضیح مراحل الگوریتم راجع به نحوه انتخاب جمعیت اولیه توضیح داده می شود. برای این منظور مفاهیم همبستگی بردارهای حرکت بررسی می گردد.

میزان جابجایی یک بلوک به جابجایی بلوکهای مجاور آن در یک فریم و همچنین به میزان جابجایی بلوک نظیرش در فریم

۲۲ بردار را تشکیل می دهند. سپس این ۲۲ بردار بر حسب تابع fitness مرتب شده، ۱۶ بردار که fitness آنها کمتر است انتخاب می شود. در حالیکه در الگوریتم PVGSA عملیات مرتب کردن انجام نمی شود و هشت کروموزم جهش یافته به همراه مقادیر هشت کروموزم قبل از جهش وارد چرخ رولت می شوند.

۴- نتایج شبیه سازی

در این بخش نتایج اعمال الگوریتم بر روی تعدادی از تصاویر ویدئویی استاندارد بررسی می شود. در شبیه سازیهای انجام شده اندازه بلوکها 16×16 و $w=7$ انتخاب شده است. تصاویر دارای فرمت CIF می باشند. در شبیه سازیها از ۳۰ فریم متوالی از این تصاویر استفاده شده است. همچنین آستانه T برابر با 64512 قرار داده شده است. مقایسه یک عدد 16 بیتی با این مقدار کار ساده ای است. زیرا 6 بیت پرارزش این عدد یک و بقیه بیتهای آن صفر است. بنابراین عددی بزرگتر از T است که 6 بیت پرارزش آن یک و حداقل یکی از دیگر بیتهای آن یک باشد که با یک گیت OR و یک گیت AND می توان چنین مقایسه ای را انجام داد. در جدول ۱ و ۲ به ترتیب مقادیر PSNR و متوسط تعداد نقاط جستجو بازای یک بلوک برای الگوریتمهای FS, 3SS, 4SS, 4GS, PVGSA مقایسه شده است.

	FS	3SS	4SS	4GS	PVGSA
Trevor	33.48	33.10	33.20	33.31	33.35
Caltrain	33.81	29.77	30.44	33.65	33.69
Miss-america	37.69	37.00	37.15	37.67	37.65
Claire	42.12	42.00	42.03	42.10	42.09
Salesman	34.76	34.64	34.68	34.66	34.58
Average	36.37	35.302	35.5	36.27	36.28

	FS	3SS	4SS	4GS	PVGSA
Trevor	208.81	22.82	16.5	34.7	17.1
Caltrain	208.81	23.84	19.1	36.3	18.2
Miss-america	208.81	22.82	19.4	36.4	16.4
Claire	208.81	23.61	16.2	35.1	16.30
Salesman	208.81	23.6	16.5	35.6	17.0

مطابق با جدول ۲، از لحاظ تعداد نقاط جستجو الگوریتم پیشنهادی 1.85 تا 2.58 برابر سریعتر از الگوریتم 4GS و 11.47 تا 12.81 برابر سریعتر از الگوریتم جستجوی کامل است. در حالیکه مطابق با جدول ۱ از لحاظ PSNR نتایج مشابه با این الگوریتمها دارد. با توجه به پیچیدگی کمتر الگوریتم PVGSA نسبت به 4GS پیاده سازی آن نیز ساده تر خواهد بود.

جمعیت اولیه. ۱۱ بردار دیگر که بطور تصادفی در اطراف بردارهای بالا تولید می شود نیز به عنوان جمعیت اولیه در نظر گرفته میشود. به این ترتیب جمعیت اولیه شامل ۱۶ بردار می باشد. از بردارهای حرکت به عنوان کروموزم استفاده می گردد.

۲- اعمال تابع fitness مطابق با رابطه ۶ روی کروموزمها.

۳- در صورتیکه بزرگترین مقدار تابع fitness که در مرحله ۲ محاسبه شد از آستانه از پیش تعریف شده T بیشتر بود الگوریتم تمام می شود. در غیر اینصورت مراحل زیر انجام می گیرد.

۴- انتخاب هشت کروموزم از بین ۱۶ کروموزم مطابق با چرخ رولت.

۵- انجام عمل جهش با توجه به رابطه ۷.

$$MV_{j,g} = MV_{j,g-1} + Pattern_n \quad (7)$$

۶- برگشت به مرحله شماره ۲ در صورتیکه تعداد دفعات اجرای الگوریتم به ۳ نرسیده باشد.

در رابطه ۷، g مشخص کننده شماره نسل می باشد به این معنا که برای کروموزمهایی که در اولین مرتبه از اجرای الگوریتم پس از مرحله شماره ۴ انتخاب می شوند مقدار g صفر است. به همین ترتیب g برای کروموزمهایی که در دومین مرتبه از اجرای الگوریتم پس از مرحله شماره ۴ انتخاب می شوند یک است. از شماره کروموزم است. مقدار n با توجه به اندیسی که کروموزم قبل از مرحله ۴ داشته بدست می آید. بطور مثال فرض کنید از بین ۱۶ بردار، بردارهای شماره ۲، ۴، ۶، ۷، ۸، ۹، ۱۱ و ۱۳ پس از مرحله شماره ۴ الگوریتم انتخاب شوند. در اینصورت سه بیت کم ارزش این اعداد یعنی ۲، ۴، ۶، ۷، ۸، ۹، ۱۰، ۱۱ و ۱۳ بجای مقدار n قرار می گیرد. اگر $g = 0$ یا $g = 1$ باشد $Pattern$ شامل بردارهای $(0,2), (0,-2), (-2,0), (2,0), (2,2), (2,-2), (-2,2), (-2,-2)$ است. همچنین $Pattern$ برای $g=2$ شامل بردارهای $(0, 1), (0, -1), (-1, 1), (-1, -1), (1, 1), (1, -1), (1, 0), (-1, 0)$ می باشد.

در این الگوریتم از crossover استفاده نشده است. زیرا مطابق با شبیه سازیهای انجام شده تاثیری در بهبود نتایج ندارد. در مقایسه با الگوریتم 4GS، نیازی به محاسبات تقسیم، باقیمانده و جمع که در رابطه ۵ معرفی شده نمی باشد و می توان بردارهای $Pattern$ را در یک جدول مانند $Lookup Table$ نگهداری کرد. همچنین در الگوریتم 4GS، پس از اعمال جهش ۱۶ بردار بدست می آید که این ۱۶ بردار به همراه بردارهای اولیه

در این مقاله روشی برای تخمین حرکت تصاویر ویدئویی بر مبنای الگوریتم ژنتیک پیشنهاد گردید. در الگوریتم پیشنهادی حداکثر طی سه مرحله بردار حرکت بدست آمد. با توجه به خاصیت همبستگی بردارهای حرکت و خاصیت ایستاد بودن حرکت، جمعیت اولیه انتخاب گردید. برای افزایش سرعت الگوریتم مقدار تابع fitness با آستانه ای از پیش تعریف شده مقایسه شد. مقایسه تابع fitness با مقدار آستانه با یک گیت AND, OR قابل تحقق بود. نتایج شبیه سازی نشان دهنده سرعت بیشتر الگوریتم پیشنهادی در مقایسه با الگوریتم مشابه بود.

مراجع

- [1] J. Lee, N. Vijaykrishnan, M.J. Irwin and W. Wolf, "An efficient architecture for motion estimation and compensation in the transform domain", *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 16, No. 2, pp. 191-201, Feb. 2006.
- [2] O. Akbulut, O. Urhan and S. Erturk, "Improved block motion estimation using block frequency warping", *IEEE Signal Processing Letter*, Vol.15, pp. 143-145, 2008.
- [3] E.D. Frimout, et al., "Parallel architecture for a pel-recursive motion estimation algorithm", *IEEE Trans. Circuits Syst. Video Technology*, Vol. 2, No. 2, pp. 159-168, Jun. 1992.
- [4] R.C. Kordasiewicz, M.D. Gallant and S. Shirani, "Affine motion prediction based on translational motion vectors", *IEEE Trans Circuits Syst. Video Technol.*, Vol. 17, No. 10, pp. 1388-1394, 2007.
- [5] L. M. Po, W. C. Ma, "A novel four-step search algorithm for fast block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no.3, pp. 313-317, Jun. 1996.
- [6] T.Koga, et al., "Motion compensated interframe coding for video conferencing," *Proc.NTC81*, New Orleans, LA, pp. C9.6.1-9.6.5, Nov. 1981
- [7] S. Zhu, K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing*, Vol. 9, No. 2, pp. 287-290, Feb. 2000.
- [8] L.P.Chau, C.Zhu, "A fast octagon based search algorithm for motion estimation", Elsevier science, *Signal Processing journal*, Vol. 83, No. 3, pp.671-675, March. 2003
- [9] G.Sorwar, M.Murshed, L.Dooley, "A fully adaptive distance-dependent thresholding search (FADTS) algorithm for performance management motion estimation," *IEEE Trans. Circuits Sys. Video Technol.*, Vol. 17, No. 4, pp. 429-440, April. 2007.
- [10] S.M.R. Soroushmehr, S. Samavi, "An adaptive block matching algorithm for motion estimation," *proceedings of the IEEE CCECE*, May 2008.
- [۱۱] محمد رضا سروش مهر، مهدی نریمانی و شادرخ سماوی، "تخمین فازی حرکت در توالی تصاویر ویدئویی" شانزدهمین کنفرانس مهندسی برق ایران، دانشگاه تربیت مدرس، اردیبهشت ۱۳۸۷.
- [12] A.M.Tourapis, O.C.Au, M.L.Liou, "Fast block-matching motion estimation using predictive motion vector field adaptive search technique (PMVFAST)," *ISO/IEC JTC1/SC29/WG11 MPEG2000/M5866*, Noordwijkerhout, The Netherlands, Mar. 2000.
- [13] I.Ahmad, W.Zheng, M.Liou, "A fast adaptive motion estimation algorithm", *IEEE Trans. Circuits Sys. Video Technol.*, Vol. 16, No. 3, pp. 420-438, March. 2006.
- [14] C.H. Lin and J.L. Wu, "A lightweight genetic bloc-matching algorithm for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 8, No. 4, pp. 386-392, Aug. 1998.
- [15] A. Wu and S. So, "VLSI implementation of genetic four-step search for block matching algorithm," *IEEE Trans. Consumer Electronics*, Vol. 49, No. 4, pp. 1474-1481, Nov. 2003.