

## تخمین حرکت در تصاویر ویدئویی با استفاده از منطق فازی

سیدمحمد رضا سیدقلعه<sup>۱\*</sup>، مهدی نریمانی<sup>۲</sup>

<sup>\*</sup> دانشکده برق و کامپیوتر، دانشگاه صنعتی اصفهان

sayedghaleh@yahoo.com

### چکیده

تخمین حرکت تکنیکی است که بطور گسترده در کاربردهای کدینگ تصاویر ویدئویی استفاده می شود. این تخمین بر مبنای تقسیم بندی فریم به بلوکهای غیر همپوشان، روشی کارآمد برای حذف افزونگی زمانی می باشد ولی پیاده سازی بلادرنگ این روش به سادگی امکان پذیر نیست. در این مقاله با استفاده از منطق فازی و در نظر گرفتن همبستگی مکانی بین بردارهای حرکت بلوکهای مجاور در یک فریم و همبستگی زمانی بین بردارهای حرکت بلوکهای نظیر هم در دو فریم متوالی، روشی سریع برای تعیین بردار حرکت بلوکها پیشنهاد می شود. روش پیشنهادی بخاطر استفاده از جداول جستجو، دارای پیچیدگی محاسباتی کم و در نتیجه پیاده سازی سخت افزاری ساده می باشد. نتایج شبیه سازی نشان دهنده کارایی روش پیشنهادی، در مقایسه با الگوریتمهای سریع موجود در این زمینه است.

### واژه‌های کلیدی: تخمین بلوکی حرکت - همبستگی بردارهای حرکت - منطق فازی

### مقدمه

داده های ویدئویی دیجیتال نیازمند حجم ذخیره سازی و پهنای باند زیاد برای انتقال می باشند. در میان همه منابع اطلاعاتی، شاید بیشترین داده های تولیدی مربوط به داده های ویدئویی باشد. بطور مثال در فرمت ویدئویی CCIR، ابعاد فریم  $720 \times 576$  پیکسل و ۲۵ فریم از تصویر در ثانیه موجود است. اگر برای هر پیکسل ۱۶ بیت در نظر گرفته شود یک ثانیه از این تصویر حجمی برابر ۱۶۶ مگابایت اشغال می کند. در صورتیکه ابعاد تصویر بزرگتر شود یا ذخیره سازی برای مدت زمان بیشتری انجام شود حجم داده ها به مراتب افزایش می یابد. فشرده سازی روشی موثر در کاهش حجم داده ها است. در فشرده سازی ویدئویی علاوه بر بهره بردن از وجود افزونگی مکانی و ویژگیهای طبیعی چشم انسان، از وجود افزونگی زمانی نیز استفاده می شود. در اینصورت امکان دستیابی به نرخهای بالای فشرده سازی امکان پذیر می گردد.

در سالهای اخیر چندین استاندارد فشرده سازی ویدئویی نظیر H.261, H.263, MPEG1/2/4 برای کاربردهای مختلف ارائه شده است. از اجزای این استانداردها بخش مربوط به کاهش افزونگی زمانی است [۱]. تخمین حرکت از جمله تکنیکهایی است که در حذف افزونگی زمانی و در نتیجه فشرده سازی بیشتر تصاویر ویدئویی کاربرد گسترده ای دارد. روشهای بسیار متنوعی برای تخمین حرکت وجود دارد. این روشها را می توان به دو دسته کلی تخمین حرکت بر مبنای انطباق بلوک و الگوریتمهای جستجوی بازگشتی پیکسل تقسیم نمود. روش مبتنی بر انطباق بلوک، روشی کارآمد در زمینه تخمین حرکت می باشد [۲]. در این روش فریم جاری به بلوکهای مربعی غیر همپوشان با ابعاد  $N \times N$  تقسیم می شود. سپس بازای هر بلوک،

۱- دانشجوی کارشناسی ارشد کامپیوتر، دانشگاه صنعتی اصفهان

۲- عضو هیأت علمی پژوهشکده علوم و تکنولوژی زیر دریا، دانشگاه صنعتی اصفهان

ناحیه ای به نام ناحیه جستجو در فریم مرجع (فریم قبل یا بعد از فریم جاری) تعیین می گردد. این ناحیه به اندازه  $w$  پیکسل از هر یک از ۴ لبه بلوک نظیر بلوک جاری در فریم مرجع گسترده می شود. برای تعیین بلوک منطبق، یک تابع معیار در نظر گرفته می شود. توابع معیار  $MAE$ ،  $MSE$  که در روابط ۱ و ۲ معرفی شده اند در میان توابع معیار بیشتر مطرح می باشند. تابع معیار  $MAE$  بخاطر پیاده سازی ساده استفاده بیشتری نسبت به  $MSE$  دارد.

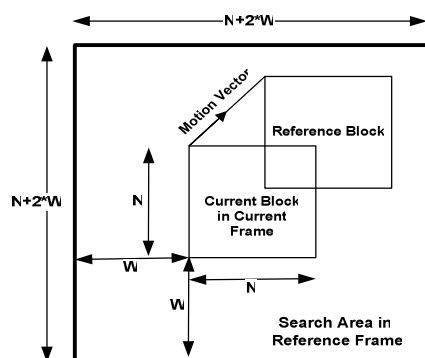
$$MAE(x, y) = (1/N^2) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} |I_{cur}(x_0 + m, y_0 + n) - I_{ref}(x_0 + x + m, y_0 + y + n)| \quad (1)$$

where  $-W \leq (x, y) \leq W$

$$MSE(x, y) = 1/N^2 \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} (I_{cur}(x_0 + m, y_0 + n) - I_{ref}(x_0 + x + m, y_0 + y + n))^2 \quad (2)$$

where  $-W \leq (x, y) \leq W$

در روابط ۱ و ۲،  $I_{ref}(i, j)$ ،  $I_{cur}(i, j)$  به ترتیب بیانگر میزان روشنایی پیکسل واقع در مکان  $(i, j)$  از فریم جاری و فریم مرجع می باشد. از بین بلوکهای این ناحیه، بلوکی که بیشترین انطباق را با بلوک جاری دارد بعنوان بلوک منطبق شناخته می شود. برداری که بین دو نقطه نظیر بلوک جاری و بلوک منطبق رسم می شود بردار حرکت نامگذاری می گردد. در شکل ۱، ناحیه جستجو به همراه بردار حرکت نمایش داده شده است.



شکل ۱- نمایش ناحیه جستجو و بردار حرکت

در الگوریتم جستجوی کامل ( $FS$ )، چون تمام بلوکهای ناحیه جستجو مورد ارزیابی قرار می گیرد نقطه می نیمم سراسری پیدا می شود. این الگوریتم درعین سادگی، نیازمند حجم محاسبات بسیار زیادی می باشد و بکار بردن آن، انجام بلادرنگ تخمین را مشکل می نماید. به این دلیل تا کنون الگوریتمهای زیادی برای افزایش سرعت جستجو و انجام بلادرنگ تخمین حرکت ارائه شده است. در برخی از الگوریتمها الگوی ثابتی برای یافتن نقطه بهینه بکار برده می شود. از جمله این الگوریتمها می توان به الگوریتم جستجوی لگاریتمی [۳]، الگوریتم جستجوی ۳ مرحله ای (3SS) [۴]، الگوریتم جستجوی چهار مرحله ای (4SS) [۵]، الگوریتم جستجو با الگوی لوزی شکل [۶] و الگوریتم جستجو با الگوی هشت وجهی [۷] اشاره نمود. در کلیه الگوریتمهای این رده فرض می شود که با دور شدن از نقطه مینیمم، میزان خطا افزایش می یابد [۳]. این فرض همیشه درست نیست. بنابراین الگوریتمهای این دسته ممکن است بجای مینیمم سراسری، مینیمم محلی را پیدا کنند. ولی می توان فرض نمود که در ناحیه کوچکی اطراف می نیمم سراسری این فرض صحیح است [۸]. بنابراین در پاره ای از الگوریتمها سعی می شود که در ابتدا بردار حرکت بلوک پیشگویی شود و سپس در اطراف مکان پیشگویی شده، جستجو انجام گیرد. اساس این پیشگوییها همبستگی بین بردارهای حرکت بلوکهای مجاور در یک فریم و همبستگی بردارهای حرکت یک بلوک با بردار حرکت بلوک نظیر آن در فریم مرجع است. در مرجع [۹]، با در نظر گرفتن همبستگی مکانی بردارهای حرکت، از منطق فازی برای پیشگویی بردار حرکت استفاده شده است. استفاده از چند جمله ایها [۱۰] نیز روشی است که برای کاهش حجم محاسبات تا کنون بکار برده شده است.

در این مقاله با در نظر گرفتن همبستگی مکانی و زمانی بردارهای حرکت، با استفاده از منطق فازی، روشی سریع برای یافتن بردارهای حرکت پیشنهاد می شود. در الگوریتم پیشنهادی، برای بلوکهای ایستان، جستجوی کمتری صورت می گیرد. در نتیجه برای تصاویری که دارای پیش زمینه زیاد یا دارای حرکات بسیار کند می باشند بردارهای حرکت بسیار سریع مشخص می شود. روند ارائه مطالب به این قرار است: پس از بخش مقدمه، مروری مختصر بر منطق فازی می شود و سپس مراحل الگوریتم پیشنهادی معرفی می شود. سپس نتایج شبیه سازی الگوریتم پیشنهادی بر روی تصاویر ویدئویی استاندارد، با الگوریتمهای موجود مقایسه می گردد. بخش نهایی این مقاله نیز به نتیجه گیری اختصاص دارد.

### منطق فازی و الگوریتم پیشنهادی *FBME* - Fuzzy Based Motion Estimation

با توجه به ویژگیهای متفاوت تصاویر ویدئویی، ارائه الگوریتمی که بتواند برای تمام توابعهای همه تصاویر، سرعت بالا و در عین حال کیفیت تصویر بازسازی شده مناسب داشته باشد کار آسانی نیست. از زمانیکه برای اولین بار پروفیسور لطفی زاده منطق فازی را پیشنهاد نمود این منطق برای کنترل سیستمهای پیچیده (Complex)، بدتعریف (ill-defined)، مبهم (vague) و ناقص (incomplete) بکار برده شده است [۹]. مزیت استفاده از این سیستمها این است که رفتار سیستم می تواند با روابط *if-then* نمایش داده شود. بطور مثال در صورتیکه دو ورودی  $X, Y$  و خروجی  $Z$  را داشته باشیم یک سیستم فازی می تواند با استفاده از قوانین زیر بیان گردد.

Rule  $i$  : if  $X$  is  $A_i$  and  $Y$  is  $B_i$  then  $Z$  is  $C_i$

مثلاً در مورد الگوریتم پیشنهادی این مقاله، یکی از قوانین فازی می تواند بصورت زیر بیان گردد. اگر جابجایی یکی از بلوکهای مجاور در راستای افقی ۲ پیکسل و جابجایی بلوک مجاور دیگر ۳ پیکسل باشد آنگاه جابجایی بلوک مورد نظر در راستای افقی حدود ۳ پیکسل خواهد شد.

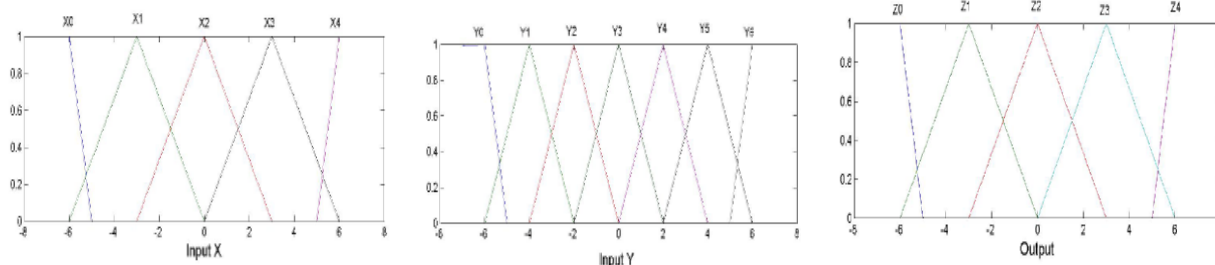
بنابراین در ابتدا باید عملیات فازی سازی انجام شود. سپس قوانین فازی روی ورودیهای فازی شده اعمال شود و میزان انطباق ورودیهای  $X, Y$  به ترتیب با  $A_i, B_i$  مشخص و از آنجا در مورد خروجی  $Z$  تصمیم گیری گردد. با توجه به اینکه خروجی  $Z$  نیز فازی است باید عملیات نافازی سازی روی آن اعمال گردد. در الگوریتم پیشنهادی این مقاله، از عملیات فازی کردن مثلثی و نافازی سازی مرکز ثقل (COG) استفاده شده است. برای پیاده سازی ساده عملیات نافازی سازی، از جدول جستجو (Lookup Table) استفاده گردیده است.

از آنجا که معمولاً یک شیء را بیش از یک بلوک دربر می گیرد حرکت بلوکهای مجاور بسیار به یکدیگر شبیه است. از این مورد به همبستگی مکانی بردارهای حرکت تعبیر می شود. با توجه به وجود اینرسی حرکتی، همبستگی زیادی بین بردارهای حرکت بلوکهای فریمهای متوالی نیز وجود دارد. یعنی اگر بلوکی از یک فریم، با بردار حرکت  $(x, y)$  جابجا شود به احتمال زیاد، بلوک نظیر آن در فریم بعد نیز با همین بردار حرکت یا با بردار حرکتی بسیار نزدیک به بردار  $(x, y)$  جابجا می شود. این ویژگی، تحت عنوان همبستگی زمانی بردارهای حرکت شناخته می شود. در الگوریتم *EFS* (Efficient Fuzzy Search) [۹]، از همبستگی مکانی بردارهای حرکت استفاده شده و بر مبنای آن، با استفاده از منطق فازی روشی برای پیشگویی نقطه شروع جستجو معرفی گردیده است. در مقاله حاضر، علاوه بر همبستگی مکانی بردارهای حرکت، از همبستگی زمانی این بردارها نیز استفاده می شود. در شکل ۲، بخشی از بلوکهای یک فریم نمایش داده شده است. 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 بلوک  $BC$ ، بردار حرکت بلوکهایی که با علامت ' مشخص شده اند معلوم می باشد. در این مقاله، از بردارهای حرکت بلوکهای  $B1, B2, B3$  و همچنین بردار حرکت بلوک نظیر  $BC$  در فریم مرجع، که با  $B4$  نامگذاری می کنیم برای پیشگویی بردار حرکت بلوک  $BC$  استفاده می گردد.

✓	✓	✓	✓
✓	B <sub>2</sub>	B <sub>1</sub>	✓
✓	B <sub>3</sub>	B <sub>C</sub>	?

شکل ۲- بلوکهای مجاور بلوک جاری

بردار حرکت بلوکهای  $B_i$  ( $1 \leq i \leq 4$ ) را با  $(MVXi, MVYi)$  نامگذاری می کنیم که  $MVXi$  و  $MVYi$  به ترتیب میزان جابجایی بلوک  $B_i$  در راستاهای افقی و عمودی نسبت به فریم مرجع است. با توجه به شبیه سازیهای انجام شده، توابع عضویت و قوانین فازی استفاده شده در مقاله پیشنهادی این مقاله بصورت شکل ۳ و جدول ۱ بیان می گردد.



شکل ۳- توابع عضویت ورودیهای X, Y و خروجی Z

جدول ۱- قوانین استفاده شده در الگوریتم پیشنهادی

	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X1	Z1	Z1	Z2	Z3	Z3	Z4	Z5
X2	Z1	Z2	Z2	Z3	Z4	Z5	Z5
X3	Z2	Z2	Z3	Z3	Z4	Z5	Z6
X4	Z3	Z3	Z4	Z4	Z5	Z5	Z6
X5	Z3	Z4	Z4	Z5	Z6	Z6	Z7

مراحل الگوریتم پیشنهادی به شرح زیر است.

- مقدار رابطه ۱ بازای  $(x,y)=(0,0)$  بدست می آید. در صورتیکه حاصل این رابطه، از آستانه از پیش تعیین شده  $T$  کمتر باشد بلوک مورد نظر ( $B_C$ )، ایستان تشخیص داده می شود. در غیر اینصورت مراحل زیر انجام می شود.
- مقدار  $(MVX2 + 7)$  به عنوان آدرس افقی و  $(MVX1 + 7)$  به عنوان آدرس عمودی جدولی با ۲۲۵ مدخل (Entry) استفاده می شود. مدخلهای این جدول، حاصل عملیات نافازی سازی است. با توجه به آدرس تعیین شده، مقداری که از جدول بدست می آید در خانه  $MVX_H$  نگه داشته می شود
- مرحله ۱، برای  $(MVX3 + 7)$  و  $(MVX2 + 7)$  و همچنین  $(MVY2 + 7)$  و  $(MVY1 + 7)$  و نیز  $(MVY3 + 7)$  و  $(MVY2 + 7)$  تکرار می شود و مقادیر نظیر این آدرسها به ترتیب در خانه های  $MVX_V$  و  $MVY_V$  و  $MVY_H$  نگهداری می گردد.

(۴) مرحله ۱، برای مقادیر  $(\frac{MVX_V + MVX_H}{2} + 7)$  و  $(\frac{MVY_V + MVY_H}{2} + 7)$  و همچنین  $(\frac{MVX_V + MVX_H}{2} + 7)$  و  $(\frac{MVY_V + MVY_H}{2} + 7)$  و  $(MVY4 + 7)$  تکرار می شود. مقادیر بدست آمده از جدول، به ترتیب در خانه های  $MVX$  و  $MVY$  ذخیره می شود.

(۵) نقطه  $(MVX, MVY)$  به عنوان نقطه شروع جستجو مشخص و پنجره  $3 \times 3$ ، به مرکز آن گسترده می گردد.  
 (۶) در صورتیکه نقطه مینیمم در مرکز پنجره بدست آید جستجو تمام می شود. در غیر اینصورت پنجره  $3 \times 3$  به مرکز نقطه مینیمم هر مرحله تا زمانی گسترده می شود که نقطه مینیمم در مرکز پنجره واقع شود یا نقاطی از پنجره به خارج مرز ناحیه جستجو برسد.

(۷) نقطه مینیمم مرحله قبل به عنوان نقطه نهایی تعیین و بردار نظیر آن به عنوان بردار حرکت شناخته می شود.

### نتایج شبیه سازی

در شبیه سازیها، از ۳۰ فریم متوالی از تصاویر ویدئویی استاندارد Claire, Susie, Garden, Trevor, Tennis با ابعاد تصویر  $352 \times 240$  و ۳۰ فریم متوالی از تصویر ویدئویی استاندارد foreman با ابعاد تصویر  $176 \times 144$  و نیز ۳۰ فریم متوالی از تصویر ویدئویی Caltrain با ابعاد تصویر  $512 \times 400$  استفاده شده است. حداکثر جابجایی (w) برابر با ۷ و ابعاد بلوکها  $16 \times 16$  فرض شده است. مقدار آستانه T برابر با ۲ انتخاب شده است. از تابع MAE به عنوان تابع معیار استفاده گردیده است. شبیه سازیها با الگوریتمهای FS, 3SS, 4SS, EFS و الگوریتم پیشنهادی FBME، برای اندازه گیری پارامترهای PSNR, MSE و میانگین تعداد نقاط جستجو برای هر بلوک انجام شده است. در جدول ۲، الگوریتمهای مختلف از نظر پارامتر PSNR برای ۳۰ فریم متوالی از تصاویر استاندارد با یکدیگر مقایسه شده اند. مطابق جدول ۲، نزدیکترین مقادیر به مقادیر الگوریتم FS، مربوط به الگوریتمهای EFS و الگوریتم پیشنهادی FBME است. تنها برای تصاویر Claire, Garden، مقادیر PSNR در الگوریتم EFS بیشتر از الگوریتم پیشنهادی است. در سایر تصاویر، مقادیر PSNR در الگوریتم پیشنهادی، از سایر الگوریتمها بیشتر می باشد. برای دو تصویر Claire, Garden نیز اختلاف PSNR الگوریتم پیشنهادی FBME با الگوریتم EFS در حد صدم دسیبل است. در جدول ۳، الگوریتمهای مختلف از نظر متوسط تعداد نقاط جستجو برای هر بلوک، با یکدیگر مورد مقایسه قرار گرفته اند. مطابق جدول ۳، کمترین تعداد نقاط جستجو، مربوط به الگوریتم پیشنهادی است. زیرا در این الگوریتم، برای بلوکهایی که ایستاد تشخیص داده می شوند تنها یک بلوک مورد ارزیابی قرار می گیرد. برای تصویر Claire که دارای حرکات کند و پیش زمینه ثابت زیاد می باشد متوسط تعداد نقاط جستجوی الگوریتم FS، حدود ۱۰۰ برابر متوسط تعداد نقاط جستجوی الگوریتم پیشنهادی است. در حالیکه برای این تصویر، اختلاف بسیار اندکی از لحاظ PSNR با الگوریتم FS مشاهده می شود. در جدول ۴، متوسط MSE برای هر پیکسل (در تصویر بازسازی شده نسبت به تصویر اصلی) برای الگوریتمهای مختلف مقایسه گردیده است. طبق جدول ۴، نزدیکترین مقادیر به مقادیر الگوریتم FS، مربوط به الگوریتمهای EFS و الگوریتم پیشنهادی FBME می باشد.

در شکل ۴-الف و ۴-ب، به ترتیب منحنی مربوط به متوسط PSNR برای ۳۰ فریم متوالی از تصویر ویدئویی Susie و Tennis برای الگوریتمهای مختلف رسم شده است. در مقیاس نشان داده شده، اختلاف چندانی بین منحنی مربوط به الگوریتمهای EFS, FBME، مشاهده نمی گردد. برای هر دو تصویر، منحنیهای مربوط به الگوریتمهای EFS و الگوریتم پیشنهادی FBME نسبت به دو الگوریتم دیگر، به منحنی الگوریتم FS نزدیکتر می باشد. برای تصویر Tennis، نزدیکترین منحنی به منحنی الگوریتم FS، مربوط به الگوریتم پیشنهادی است.

جدول ۲- متوسط PSNR برای الگوریتم‌های معرفی شده، با الگوریتم پیشنهادی FBME

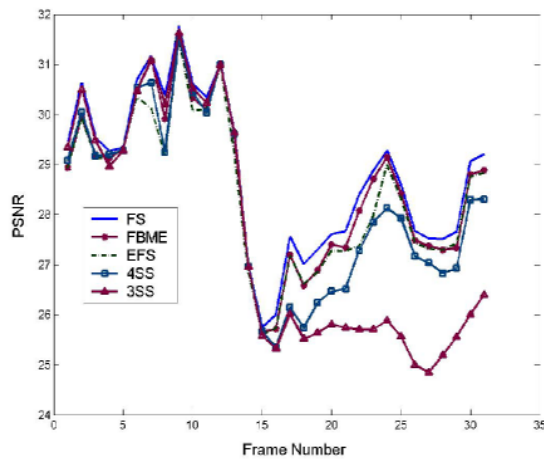
	FS	3SS	4SS	EFS	FBME
Claire	42.12	42.0	42.03	42.10	42.09
Susie	35.31	33.92	34.26	34.93	35.02
Garden	23.85	22.99	23.34	23.71	23.70
Trevor	33.56	33.09	33.20	33.48	33.51
Caltrain	30.86	29.77	30.44	30.74	30.75
Foreman	28.06	27.65	27.71	27.64	27.83
Tennis	28.82	27.57	28.21	28.42	28.57
Average	31.79	30.99	31.31	31.57	31.64

جدول ۳- متوسط تعداد نقاط جستجو برای الگوریتم‌های معرفی شده، با الگوریتم پیشنهادی FBME

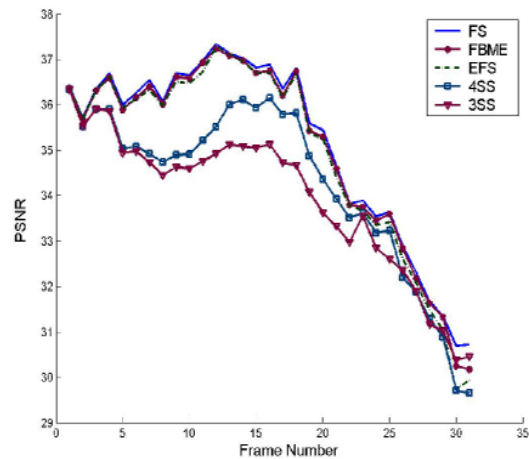
	FS	3SS	4SS	EFS	FBME
Claire	202.05	23.61	16.18	9.09	2.02
Susie	202.05	23.14	18.04	12.75	9.27
Garden	202.05	23.17	19.06	12.19	11.11
Trevor	202.05	22.82	16.54	11.17	8.73
Caltrain	210.28	23.84	19.09	13.08	12.50
Foreman	184.56	21.86	18.49	15.60	14.61
Tennis	202.05	23.15	17.89	11.70	11.07

جدول ۴- متوسط MSE (بازای هر پیکسل) برای الگوریتم‌های معرفی شده، با الگوریتم پیشنهادی FBME

	FS	3SS	4SS	EFS	FBME
Claire	4.21	4.34	4.29	4.22	4.23
Susie	21.32	28.38	27.16	24.18	23.38
Garden	270.64	330.18	305.0	279.39	279.63
Trevor	29.75	33.11	32.30	30.36	30.23
Caltrain	54.41	69.17	59.51	55.84	55.71
Foreman	109.49	120.10	118.71	122.05	116.58
Tennis	90.74	129.03	105.89	99.06	96.10
Average	82.94	102.04	93.27	87.87	86.55



شکل ۴-ب - منحنی PSNR برای ۳۰ فریم متوالی از تصویر Tennis



شکل ۴-الف - منحنی PSNR برای ۳۰ فریم متوالی از تصویر Susie

### نتیجه گیری

در این مقاله، روشی سریع برای یافتن بردارهای حرکت در الگوریتم تخمین حرکت پیشنهاد شد. در الگوریتم پیشنهادی، از همبستگی بین بردارهای حرکت بلوکهای مجاور در یک فریم و فریمهای متوالی استفاده گردید و با استفاده از منطق فازی، بردار حرکت بلوک پیشگویی شد. با توجه به اینکه عملیات نافازی سازی نیاز به محاسبات پیچیده دارد استفاده از جداول جستجو، روشی بود که برای این بخش مورد استفاده قرار گرفت. در الگوریتم پیشنهادی، برای تصاویری که دارای پیش زمینه یا حرکات کند زیادی بودند بدون کاهش محسوس کیفیت تصویر بازسازی شده، جستجوی کمتری صورت گرفت. در شبیه سازیها، برای چنین تصویری تعداد نقاط جستجو حدود ۱ درصد الگوریتم جستجوی کامل بود در حالیکه PSNR، حدود ۰/۰۳ دسیبل کمتر از الگوریتم جستجوی کامل بدست آمد.

### مراجع

- [1] Ghanbari, M., Standard codecs: image compression to advanced video coding, IEE Series, London, 2003.
- [2] Nam, J.Y., et al, "New fast search algorithm for block matching motion estimation using temporal and spatial correlation of motion vector", IEEE Transaction on Consumer Electronics, Vol. 46, No. 4, Pages. 934-942, Nov. 2000.
- [3] Jain, J., Jain, A., "Displacement measurement and its application in interframe image coding", IEEE Transaction on Communication, Vol.COMM-29, Pages. 1799-1808, Dec. 1981.
- [4] Koga, T., et al., "Motion compensated interframe coding for video conferencing", Proc.NTC81, New Orleans, LA, Pages. C9.6.1-9.6.5, Nov. 1981.
- [5] Po, L.M., Ma, W.C., "A novel four-step search algorithm for fast block motion estimation", IEEE Transaction on Circuits and System for Video Technology., Vol. 6, No. 3, Pages.313-317, June. 1996.
- [6] Tham, J.Y., et al, "A novel unrestricted center-biased diamond search algorithm for block motion estimation", IEEE Transaction on Circuits and System for Video Technology, Vol. 8, No. 4, Pages. 369-377, Aug. 1998.
- [7] Chau, L.P., Zhu, C., "A fast octagon based search algorithm for motion estimation", Elsevier science, Signal Processing journal , Vol. 83, No. 3, Pages.671-675, March. 2003.
- [8] Zeng, B., Li, R., Liou, M.L., "Optimization of fast block motion estimation algorithms", IEEE Transaction on Circuits and System for Video Technology, Vol. 7, No. 6, Pages. 833-844, Dec. 1997.
- [9] Chen, P.Y., Jou, J.M., "An efficient blocking matching algorithm based on fuzzy reasoning", IEEE Transaction on system, Man, Cybernetics.-Part B, Vol. 31, No. 2, Pages. 253-259, Apr. 2001.
- [10] Kuo, C.J., et al, "Polynomial search algorithms for motion estimation", IEEE Transaction on Circuits and System for Video Technology, Vol. 10, No. 5, Pages. 813-818, Aug. 2000.