بهبود الگوریتم تخمین حرکت WMBPM و ارائهٔ یک معماری کارآمد برای پیاده‌سازی سخت‌افزاری آن

مجتبی کرمی[۱]، مهدی سعیدی[۲]، مرتضی صاحب‌الزمانی[۳]، محمد رحمتی[۴]

چکیده

تخمین بردار حرکت یکی از بحرانی‌ترین وظایف در سیستم‌های فشرده‌سازی ویدئو می‌باشد و به همین دلیل روشهای متفاوتی برای افزایش سرعت آن ارائه شده است. این مقاله به ارائهٔ یک معماری مبتنی بر سخت‌افزار برای پیاده‌سازی الگوریتم‌های تخمین بردار حرکت می‌پردازد. به منظور ارائهٔ یک معماری کارآمد، ابتدا الگوریتم WMBPM با توجه به هزینهٔ پیاده‌سازی سخت‌افزاری بهبود داده شده است و سپس یک معماری کارآمد برای الگوریتم بهبود داده شده ارائه و ارزیابی شده است. نتایج آزمایشات انجام شده نشان می‌دهد که الگوریتم بهبود داده شده کارآمدتر از روش اولیه عمل می‌کند.

کلمات کلیدی

فشرده‌سازی ویدئو، تخمین حرکت.

Improved the WMBPM Motion Estimation Algorithm with its Efficient Hardware Architecture

Mojtaba Karami, Mehdi Saeedi, Morteza Saheb Zamani, Mohammad Rahmati

Abstract

Motion vector estimation is a critical task in video compression systems and therefore, different methods are used to increase its processing speed. In this paper, a hardware-based architecture for this component is proposed. In order to present an efficient architecture, we first improve the WMBPM algorithm by considering its implementation cost. Then, an efficient architecture for the improved WMBPM algorithm is presented and evaluated. The experimental results show that the proposed method works more efficiently than the previous algorithm.

Keywords

Video Compression, Motion Vector Estimation.

[1] دانشجوی دکترا، دانشکدهٔ مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، پست الکترونیک: m.karami@aut.ac.ir
[2] دانشجوی دکترا، دانشکدهٔ مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، پست الکترونیک: saeedi@ce.aut.ac.ir
[3] دکترا، عضو هیأت علمی، دانشکدهٔ مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، پست الکترونیک: szamani@ce.aut.ac.ir
[4] دکترا، عضو هیأت علمی، دانشکدهٔ مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، پست الکترونیک: rahmati@aut.ac.ir

۱- مقدمه

استفاده از تکنیک‌های فشرده‌سازی برای ذخیره‌سازی و ارسال تصاویر ویدئو همواره مورد توجه بوده است [1]، [2] و از این‌رو بهبود روشهای بکارگرفته در تخمین حرکت بسیار مفید است. روش انطباق بلوک[1] (BM) یکی از تکنیک‌های ساده‌ای است که در بخش تخمین حرکت بکار گرفته شده است [2]. به دلیل کاربرد وسیع الگوریتمهای الگوریتم‌های انطباق بلوک[2] (BMA)، این مقاله به بهبود الگوریتم WMBMP[3] و نیز ارائهٔ یک معماری سخت‌افزاری کارآمد برای آن تمرکز یافته است.

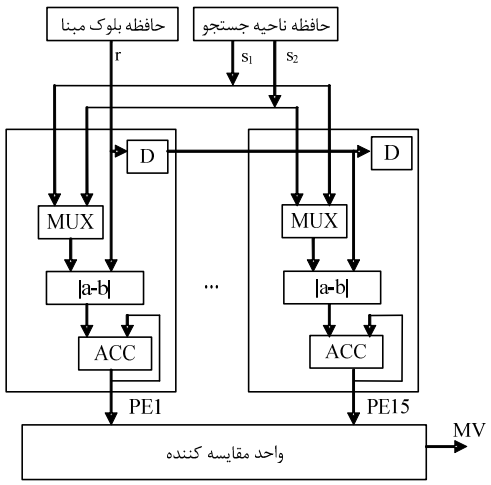
۲- معرفی الگوریتمهای انطباق صفحهٔ بیت

بهترین معیار برای انطباق مجموع مربعات خطا [4] (SSD) می‌باشد [5] که به دلیل بار محاسبات زیاد، در بسیاری از موارد به جای آن از مجموع قدرمطلق‌های خطا [5] (SAD) استفاده می‌شود [5]. در هر صورت یکی از راه‌های کاهش پیچیدگی سخت‌افزاری تخمین حرکت، استفاده از تعداد بیت کمتر برای مقایسه است. بر این اساس ساده‌ترین روش، استفاده از تنها یک بیت برای هر پیکسل می‌باشد. اما به دلیل تفاوت در اطلاعات داخل یک فریم و نیز تغییرات بین فریم‌ها در رشتهٔ تصاویر مختلف در کاربردهای گوناگون، انتخاب بیت یکسان همواره بهترین نتیجه را نمی‌دهد. برای رفع این مشکل، در مقالهٔ [۹] روشهای MPBM[6] و WMBPM بر اساس انطباق صفحهٔ بیت ارائه شده است.

۳- پیاده‌سازی سخت‌افزاری

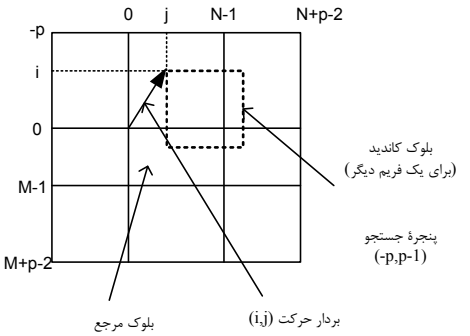
۳-۱- معماری عمومی روش تخمین حرکت

شکل ۱ معماری عمومی پیاده‌سازی معیار انطباق SAD با استفاده از ۱۶ واحد پردازشی را نشان می‌دهد.

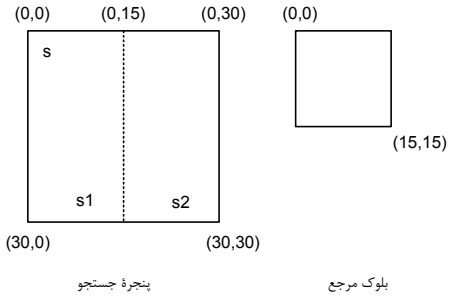


شکل ۱- معماری آرایه‌های خطی مبتنی بر بردار حرکت برای تخمین حرکت با ۱۶ عنصر پردازشی [2]

شکل ۲-الف یک پنجرهٔ جستجوی ۱۶×۱۶ پیکسل را با در نظر گرفتن یک محدوده جستجوی [-p,p-1] با p=8 نشان می‌دهد. در شکل ۲-ب یک پنجرهٔ جستجوی ۳۱×۳۱ برای معماری آرایه‌ای خطی و بلوک مبنا نشان داده شده است. هر واحد در شکل ۱ یک مقدار *BPM(i,j)* را محاسبه می‌کند. ورودی *s1* داده‌ها را از نیمهٔ چپ و ورودی *s2* داده‌ها را از نیمهٔ راست پنجره جستجو دریافت می‌کند (شکل ۲). این ورودیها به همهٔ المانهای پردازشی وارد می‌شوند. پس از محاسبهٔ ۱۶ مقدار *BPM(i,j)* برای سطر اول پنجرهٔ جستجو (از *i=0, j=0* تا *i=0, j=15*)، واحد مقایسه‌گر فعال شده و مینیمم تعیین می‌شود. سپس مقادیر *BPM(i,j)* برای سطر دوم پنجرهٔ جستجو محاسبه و واحد مقایسه‌گر فعال و مینیمم بروز می‌شود. این‌کار ۱۶ بار تکرار شده و مینیمم *BPM(i,j)* در پنجرهٔ جستجو بدست می‌آید. هنگام شروع محاسبه برای یک سطر جدید در پنجره جستجو، بلوک مبنا دوباره از اول به واحد اول پردازشی داده می‌شود.



الف



ب

شکل ۲-الف) پنجره جستجو در تخمین حرکت با محدوده [-p,p-1]
ب) بلوک مبنا ۱۶×۱۶ و پنجره جستجو ۳۱×۳۱ [2]

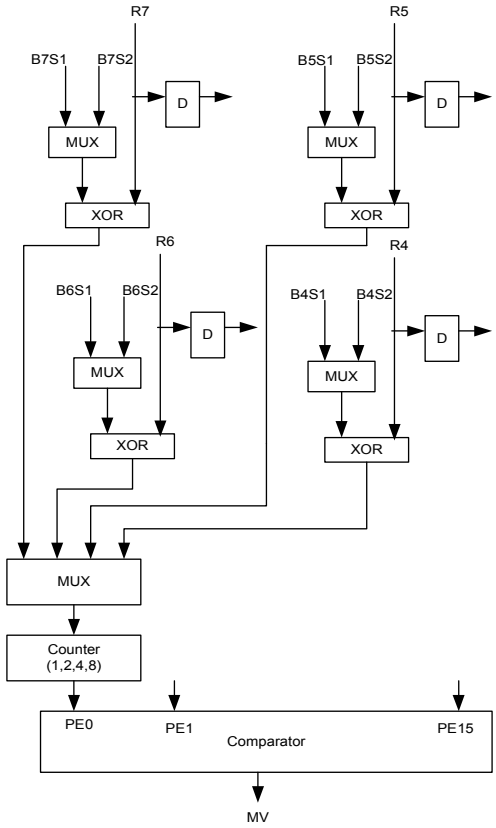
۳-۲- تخمین حرکت به روش WMBPM

معماری الگوریتم تخمین حرکت WMBPM در شکل ۳ نشان داده شده است.

۳-۳- بهبود روش WMBPM و یک معماری جدید

در روش WMBPM، در هر زمان تنها یکی از ورودیها معتبر است. به عبارت دیگر با توجه به ورودی معتبر مالتی‌پلکسر چهار ورودی، نتیجه

به یک شمارنده وارد و در صورت یک بودن نتیجه و بر اساس اینکه کدام یک از ورودیهای R4 تا R7 در آن زمان معتبر باشد، نتیجهٔ موجود در شمارنده با ۱، ۲، ۴ یا ۸ جمع می‌شود.

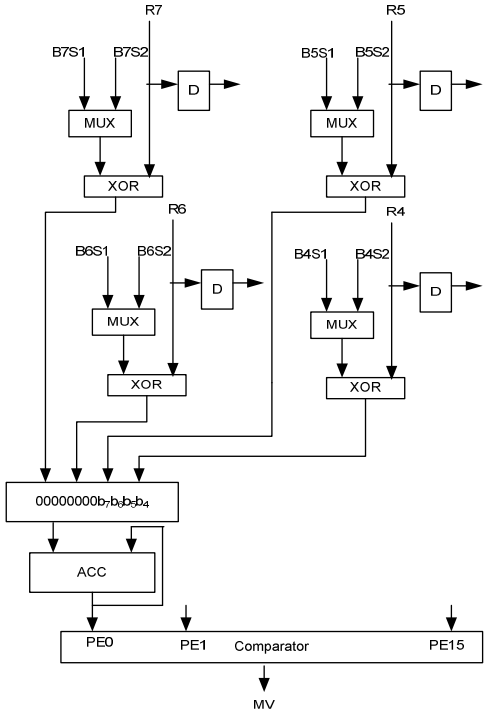


شکل ۳ معماری تخمین حرکت به روش WMBPM

با قرار دادن خروجی XOR ها در کنار یکدیگر، می‌توان آنها را بصورت یک عدد در نظر گرفت. به این ترتیب با توجه به اینکه خروجی XOR مربوط به R4 در مکان بیت صفر قرار گرفته است، اگر این خروجی فعال باشد در عدد حاصل بیت صفر برابر با یک است و این مسأله به معنای جمع با یک می‌باشد. به همین ترتیب خروجی XOR مربوط به R5 در مکان بیت اول (جمع با ۲)، خروجی XOR مربوط به R6 در مکان بیت دوم (جمع با ۴) و خروجی XOR مربوط به R7 در مکان بیت سوم (جمع با ۸) قرار گرفته‌اند. با این تغییر وجود مالتی‌پلکسر چهار ورودی حذف شده و به جای یک شمارنده با قابلیت جمع با ۱، ۲، ۴ و ۸، یک انباشتگر قرار داده می‌شود که همواره عمل جمع ورودی با مقدار قبلی را انجام خواهد داد. به این ترتیب مالتی‌پلکسر چهار ورودی و مدار کنترل لازم برای انتخاب ورودی آن و نیز انتخاب نوع جمع در شمارنده مورد نیاز نبوده و حجم سخت‌افزار استفاده شده بسیار کاهش و سرعت نیز افزایش می‌یابد(شکل ۴).

۳-۴- نتایج پیاده‌سازی

معماری ارائه شده بر روی FPGA هدف v50bg256-6 پیاده‌سازی شده است. فرکانس بیشینهٔ قابل استفاده در این پیاده‌سازی ۷۴ مگاهرتز گزارش شده است. بعلاوه برای تخمین بردارهای حرکت یک فریم به اندازهٔ ۴۱۱۱ سیکل ساعت زمان نیاز خواهد بود که ناشی از آماده شدن نتایج کل عناصر پردازشی و سپس انتخاب بهترین گزینه است. با توجه به فرکانس گزارش شده و نیز نظر به کاربردهای معمولی تصاویر ویدئویی که حداکثر دارای سرعت ۳۰ فریم در ثانیه می‌باشند، می‌توان نشان داد که معماری ارائه شده به راحتی توانایی تخمین حرکت موجود در تصاویر ویدئویی را داراست. بعلاوه حداکثر توانایی تخمین حرکت این معماری برابر با $74.355M/4111 \approx 18000\ fps$ می‌باشد و بنابراین این معماری برای کاربردهای ویژه‌ای که دارای سرعت فریم در ثانیه بسیار زیاد می‌باشند نیز کاملاً مناسب است.



شکل ۴ معماری بهبود یافته تخمین حرکت به روش WMBPM

۴- بهبود الگوریتم WMBPM

معماری ارائه شده در شکل ۳، توسط پوررضا و همکارانش [۹] پیشنهاد شده است این معماری با هدف کاهش پیچیدگی سخت‌افزاری بصورت شکل ۴ تصحیح شد. با نگاهی دوباره به این معماری می‌توان مشاهده کرد که در ورودی انباشتگر در هر واحد پردازشی، چهار بیت حاصل از نتیجه XOR فریمهای مبنا و مرجع در کنار یکدیگر قرار گرفته‌اند. از طرفی با توجه به روابط پیشین، تنها یک مقدار از چهار مقدار یک پیکسل در چهار صفحهٔ بیت، برابر با ۱ بوده و سه مقدار دیگر دارای ارزش صفر هستند. بنابراین می‌توان از مقادیر موجود در چهار صفحه بیت بطور همزمان استفاده کرد. مقدار صفحات بیت بدون توجه به مکان پیکسل، می‌تواند به صورت زیر باشد:

$$T_i[\ f(i,j)] = b_i \qquad (1)$$

به این ترتیب برای هر پیکسل به جای اطلاعات موجود در یک بیت، اطلاعات هر چهار بیت مورد استفاده قرار می‌گیرد. باید توجه کرد

که این بهبود در الگوریتم، افزایشی در مدارات سخت‌افزار را در بر نخواهد داشت. از طرفی با توجه به اینکه در رابطهٔ ۱ نیازی به محاسبه برای تعیین صفحات بیت نیست، این روش در تعیین صفحات بیت سریعتر از روش WMBPM بوده و کارایی بهتری نیز دارد.

برای ارزیابی بهتر، دو روش WMBPM و بهبود یافتهٔ آن بر روی شش رشتهٔ ۳۰۰ فریمی که در آزمایشهای فشرده‌سازی تصاویر ویدیویی، بسیار مورد استفاده قرار می‌گیرند نیز ارزیابی شده‌اند (شکل ۵). برای مقایسهٔ بهتر و ارزیابی آماری، از معیار PSNR [۹] با تعریف زیر استفاده شده است:

$$PSNR = 10\log_{10}\left(\frac{255^2}{\frac{1}{X.Y}\sum_{x=1}^{X}\sum_{y=1}^{Y}(C(x,y)-\tilde{C}(x,y))^2}\right) \quad (2)$$

که در این رابطه، $\hat{C}(x,y)$ تخمین $C(x,y)$ با استفاده از بردارهای حرکت بدست آمده است. نتایج بدست آمده در جدول ۱ نشان داده شده است. همانطوری‌که در این جدول نیز مشخص است، روش بهبود داده شده ۳/۵ درصد بهتر از روش اولیه در تخمین حرکت کارایی دارد.

۵- نتیجه‌گیری

در این مقاله معماری سخت‌افزاری دو روش تخمین حرکت مبتنی بر تطبیق بلوک ارائه گردید. سپس یک معماری بهبود یافته برای الگوریتم WMBPM ارائه و ارزیابی شد. در ادامه بر اساس معماری ارائه شده به بهبود الگوریتم WMBPM پرداخته و نشان دادیم که با استفاده از الگوریتم بهبود داده شده می‌توان بدون افزایش در پیچیدگی سخت‌افزاری تا ۳/۵ درصد بهتر از روش اولیه به تخمین حرکت پرداخت که این امر به آن معنا است که مقدار بهبود نسبت به الگوریتم مرجع به میزان ۲ برابر ارتقا یافته است.

مراجع

[1] P.M. Kuhn, G. Diebel, S. Herrmann, A. Keil, H. Mooshofer, A. Kaup, R. Mayer, W. Stechele, "*Complexity and PSNR-Comparison of several Fast Motion Estimation Algorithms for MPEG-4,*" SPIE, 1998.

[2] V. Bhaskaran and K. Konstantinides, *"Image and Video Compression Standards,"* Kluwer Academic Publishers, 1997.

[3] S.J. Ko, S.H. Lee, and K.H. Lee, *"Digital Image Stabilizing Algorithms Based on Bit-Plane Matching,"* IEEE Trans. On Consumer Electronics, Vol. 44, No. 3, pp. 617-622, 1998.

[4] J. Zhang, O.M. Ahmad, M.N.S. Swamy, *"Feature-Bit-Plane Matching Technique for Estimation of Motion Vectors,"* Electronic Letters, Vol. 34, No. 11, pp. 1090-1091, 1998.

[5] N. Sebe, M.S. Lew, D.P. Huijsmans, *"Toward Improved Ranking Metrics,"* IEEE Trans on PAMI; Vol. 22; No. 10; pp. 1132-1143; 2000.

[6] J. Feng, K.-T. Lo, H. Mehrpour and A.E. Karbowiak, *"Adaptive Block Matching Motion Estimation Algorithm Using Bit-Plane Matching,"* Proc. Int. Conf. Image Processing, 1995.

[7] H.R. Pourreza, M. Rahmati, F. Behazin, *"Weighted Multiple Bit-Plane Matching, a Simple and Efficient Matching Criterion for Electronic Digital Image Stabilizer Application,"* Proc. Int. Conf. on Signal Processing, 2002.

[8] H.R. Pourreza, M. Rahmati, F. Behazin, *"Adaptive Pixel Difference Classification, an Efficient and Cost Effective Algorithm for Motion Estimation,"* Proc. of 2nd WSEAS Int. Conf. On Signal, Speech and Image Processing (WSEAS ICOSSIP'02), 2002.

[۹] حمیدرضا پوررضا، فرید به‌آذین، محمد رحمتی، "چند معیار جدید انطباق، مناسب پیاده‌سازی سخت‌افزاری برای کاربرد فشرده‌سازی ویدیو" امیرکبیر، سال پانزدهم، شماره آ-۵۷ (مهندسی برق) زمستان ۱۳۸۲.

شکل ۵- فریم اول رشتهٔ تصاویر آزمایشی، از چپ به راست و از بالا به پایین: Akiyo، Coastguard، Container، Hall، Mother، Silent

جدول ۱- نتایج آزمایش الگوریتم WMBPM و بهبودیافته آن

رشته فریم	WMBPM	الگوریتم بهبود داده شده
akiyo	44.6253	45.4362
coastguard	33.6834	37.7573
container	44.2331	44.8173
hall	41.1475	41.7648
mother	42.5356	43.8780
silent	40.6080	41.7049
میانگین	41.1388	42.5597

زیرنویس‌ها

[1] Block Matching
[2] Block Matching Algorithm
[3] Weighted Multiple Bit-Plane Matching
[4] Sum of Square Difference
[5] Sum of Absolute Difference
[6] Multiple Bit-Plane Matching