

پیاده‌سازی شار نوری هرن و شانک بر روی FPGA

امین بهرامی^۱، محمدرضا محزون^۲، احمدرضا امین^۲، محمد رحیمی^۳، جواد عرب^۳

تاریخ دریافت: ۹۱/۰۳/۲۳

تاریخ پذیرش: ۹۱/۰۵/۰۸

چکیده

محاسبه سرعت کلیه نقاط متحرک، از موضوعات مهم پردازش تصویر می‌باشد. روش‌های موسوم به «استخراج بردارهای شار نوری از دنباله تصاویر» در این راستا مطرح می‌باشند. کاربردهای مختلف شار نوری در امور نظامی از جمله: رهگیری هدف، PIV، یافتن عمق تصویر، کنترل هواپیماهای بدون سرنشین، مسافت‌سنجی، کنترل ربات و... بر هیچ کس پوشیده نیست. با توجه به ماهیت غیر فعال بودن سیستم تصویربرداری و محاسبه شار تصاویر، به‌کارگیری چنین سیستمی در کاربردهای پدافند غیرعامل مناسب خواهد بود. اگر چه تا به حال، روش‌های موفق و متعدد شار نوری مطرح شده‌اند، لیکن پیاده‌سازی این روش‌ها به لحاظ بار محاسباتی و کند شدن پاسخ سیستم به‌کارگیرنده معمولاً به صرفه نبوده است. انتخاب سخت‌افزاری سریع و همچنین انتخاب الگوریتمی مناسب که قابل پیاده‌سازی بر سخت‌افزار مذکور بوده و منجر به تولید یک سیستم استخراج بردارهای شار نوری صحنه باشد موضوع مقاله حاضر است. الگوریتم مورد استفاده در این رابطه، روش شار نوری هرن و شانک و سخت‌افزار انتخابی FPGA نوع Spartan 3A DSP3400 محصول شرکت Xilinx می‌باشد. در این مقاله پس از پیاده‌سازی شار نوری هرن و شانک نتایج حاصل از پیاده‌سازی با تصاویر استاندارد، تست و مورد بررسی قرار می‌گیرند.

کلیدواژه‌ها: شار نوری، تراشه‌های برنامه‌پذیر، تعیین سرعت و پردازش تصویر

۱- دانش‌آموخته کارشناسی ارشد الکترونیک- دانشگاه جامع امام حسین(ع) bahrami.amin@yahoo.com - نویسنده مسئول

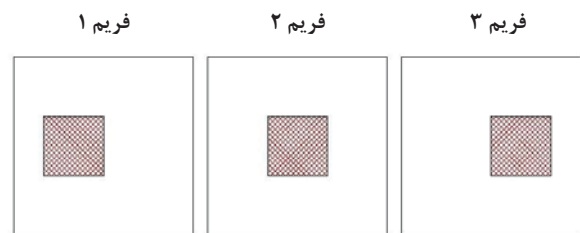
۲- استادیار و عضو هیئت علمی دانشکده و پژوهشکده فناوری اطلاعات و ارتباطات- دانشگاه جامع امام حسین(ع)

۳- دانش‌آموخته کارشناسی ارشد الکترونیک- دانشگاه جامع امام حسین(ع)

مقدمه

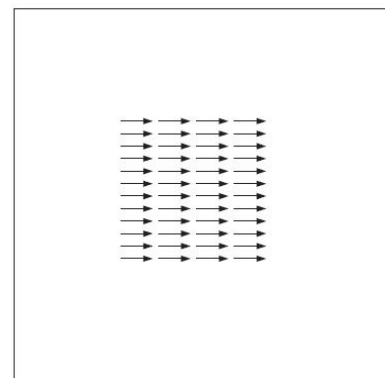
یکی از موضوعات مهم در مباحث پدافند غیرعامل، دستیابی به اطلاعات محیط اطراف بدون هیچ‌گونه پرتوزدایی جهت به‌کارگیری اطلاعات مذکور در کاربردهای مختلف از جمله رهگیری هدف، ناوبری و... می‌باشد. استفاده از سنسورهای تصویربرداری به‌عنوان عنصری غیرفعال و اطلاعات موجود در تصاویر احذ شده از جمله راه‌کارهای پیشنهادی در این زمینه است.

یکی از اصلی‌ترین روش‌های استخراج اطلاعات مربوط به دنباله‌ای از تصاویر، تعیین سرعت و حرکت اجسام حاضر در صحنه تصویربرداری است. الگوریتم‌های متعددی توسط افراد مختلف برای این منظور ارائه شده‌اند که هر کدام سعی کرده‌اند روش‌های پیشین را بهبود دهند [۳ و ۲ و ۱]. یکی از معمول‌ترین روش‌ها برای تخمین حرکت در تصاویر دو بعدی استفاده از شار نوری است. شار نوری عبارت است از حرکت الگوی روشنایی در دنباله‌ای از تصاویر که می‌تواند تقریبی از میزان حرکت هر پیکسل یا گروهی از پیکسل‌های موجود در یک تصویر باشد. شار نوری در واقع یک میدان برداری برای نشان دادن سرعت و جهت حرکت اجسام موجود در تصویر است. شکل (۱) حرکت گروهی از پیکسل‌ها به سمت راست را در سه فریم متوالی نمایش می‌دهد.



شکل ۱- نمایش حرکت گروهی پیکسل‌ها به سمت راست

شکل (۲) شار نوری محاسبه شده برای فریم دوم از شکل (۱) را نمایش می‌دهد.



شکل ۲- نمایش شار نوری برای فریم میانی شکل ۱

همان‌گونه که شکل (۲) نمایش می‌دهد، میدان برداری شار نوری ابزار مناسبی برای تشخیص جهت و اندازه حرکت اجسام متحرک در تصویر می‌باشد. شار نوری بر اساس تکنیک‌های مختلفی قابل پیاده‌سازی است که از میان آن‌ها تکنیک‌های مبتنی بر گرادیان بیشتر از سایر تکنیک‌ها در کاربردهای مربوط به سیستم‌های بلادرنگ مورد استفاده قرار می‌گیرند [۴ و ۵ و ۶]. علی‌رغم پیشرفت روزافزون رایانه‌ها، هنوز امکان پیاده‌سازی بلادرنگ شار نوری به دلیل پیچیدگی محاسباتی بر روی این سیستم‌ها وجود ندارد. به‌دلیل امکان پیاده‌سازی بخش‌های مستقل یک الگوریتم بر روی تراشه‌های برنامه‌پذیر، امکان محاسبه پاسخ سیستم در مدت زمان بسیار کمتری توسط این تراشه‌ها نسبت به رایانه وجود دارد. معیارهای برتری الگوریتم‌های محاسبه شار نوری، سرعت و دقت محاسبه شار نوری می‌باشند [۷]. در ادامه پس از معرفی الگوریتم‌های محاسبه شار نوری هرن - شانک، نحوه پیاده‌سازی این الگوریتم بیان می‌شود. در نهایت نیز نتایج حاصل از پیاده‌سازی با شار صحیح برای تصاویر استاندارد ارائه شده توسط بارن^۱ و همکارانش مقایسه می‌شوند.

الگوریتم هرن و شانک برای محاسبه شار نوری

اگر شدت روشنایی تصویر در نقطه (x, y) و در لحظه t با $E(x, y, t)$ نشان داده شود می‌توان ثابت کرد که [۹]:

$$E_x u + E_y v + E_t = 0 \quad (1)$$

که در آن، E_x ، E_y و E_t به ترتیب مشتق‌های جزئی نسبت به x ، y و t می‌باشند. معادله فوق موسوم به معادله شرطی^۲ شار نوری است. معادله فوق دو متغیر به نام‌های u و v دارد که به ترتیب بیانگر سرعت در راستای x و y برای هر پیکسل می‌باشند و قابل محاسبه از تک معادله فوق نیستند. هر پیکسل دارای سرعت در راستای x و y است لذا انتظار می‌رود که هر یک از ماتریس‌های مربوط به u و v ، بعدی برابر با بعد تصویر داشته باشند. هرن و شانک با در نظر گرفتن شروط اضافی مانند شرط یکنواختی^۳ توانستند سرعت در راستای x و y را با استفاده از روابط تکراری^۲ و ۳ محاسبه کنند [۹].

$$u^{n+1} = \bar{u}^n - E_x [E_x \bar{u}^n + E_y \bar{v}^n + E_t] \quad (2)$$

$$/ (\alpha^2 + E_y^2 + E_x^2)$$

$$v^{n+1} = \bar{v}^n - E_y [E_x \bar{u}^n + E_y \bar{v}^n + E_t] \quad (3)$$

$$/ (\alpha^2 + E_y^2 + E_x^2)$$

1- Barron et al.

2- Constraint Equation

3- smoothness

الگوریتم می‌توانند به صورت مستقل و همزمان صورت پذیرند. همانگونه که در شکل (۴) ملاحظه می‌شود محاسبات مشتق‌های تصویر و لاپلاسیان سرعت‌های u و v نیز در الگوریتم شار نوری هرن و شانک، می‌توانند به صورت کاملاً مستقل از یکدیگر و به‌طور همزمان اجرا شوند، لذا بهترین روش برای پیاده‌سازی شار در تراشه‌های برنامه‌ریزی، روشی است که محاسبات مستقل آن به صورت موازی انجام شوند، در این مقاله نیز مشتق‌های تصویر و محاسبات لاپلاسیان به‌صورت کاملاً موازی انجام شده‌اند. پس از محاسبه مشتقات تصویر و لاپلاسیان سرعت‌های u و v نهایی انجام می‌شوند. لازم به ذکر است که در این مقاله از برد آزمایشگاهی Spartan3A DSP 3400 Starter Kit که شامل تراشه XC3SD3400A می‌باشد برای پیاده‌سازی شار نوری استفاده شده است. این تراشه دارای امکانات زیادی از جمله ۱۲۶ بلوک ضرب‌کننده و ۲۸۳ کیلوبایت حافظه داخلی و... می‌باشد که استفاده از آن را برای سیستم‌های مبتنی بر پردازش تصویر مناسب کرده است [۱۰].



شکل ۳- ضرایب به‌کار رفته برای محاسبه میانگین سرعت پیکسل

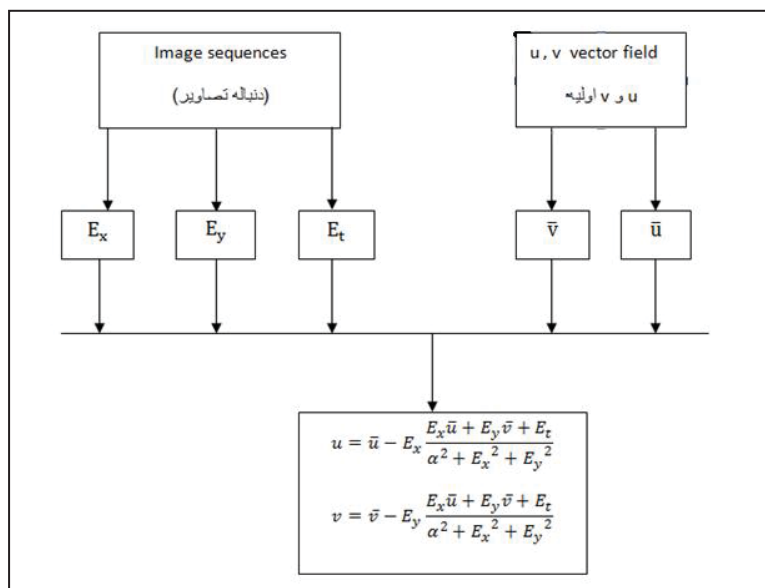
در روابط فوق u^{n+1} و v^{n+1} سرعت در تکرار $n+1$ است. \bar{u}^n و \bar{v}^n نیز میانگین سرعت پیکسل است که در مرحله n و با کمک لاپلاسیان بردارهای u و v به دست آمده‌اند. در روابط فوق، n تعداد تکرار برای حل معادله و رسیدن به یک پاسخ مناسب است. تناظر خاصی بین فیلترینگ حوزه زمان و حوزه فرکانس در مباحث پردازش تصویر وجود دارد. به دست آوردن تبدیل فوریه معکوس (IDFT) فیلترهای حوزه فرکانس، یک نقاب را برای فیلتر حوزه مکان نتیجه می‌دهد [۸]. لاپلاسیان نیز در واقع یک فیلتر حوزه فرکانس است که می‌تواند به صورت یک نقاب در حوزه مکان در نظر گرفته شود. لذا با استفاده از تئوری پیچش در مباحث پردازش تصویر می‌توان لاپلاسیان یک سیگنال گسسته دوبعدی (ماتریس‌های u و v که بعدی برابر با بعد تصویر دارند) را با پیچش یک نقاب خاص بر آن سیگنال گسسته به‌دست آورد. هرن و شانک نیز از یک نقاب خاص برای تخمین لاپلاسیان بهره بردند و با استفاده از لاپلاسیان حاصل، میانگین سرعت پیکسل (\bar{u}^n و \bar{v}^n) را با استفاده از پنجره نشان داده شده در شکل (۳) تخمین زدند. u و v در حالت اولیه ماتریس صفر در نظر گرفته می‌شوند.

E_x ، E_y ، E_t نیز با استفاده از فیلترهای 2×2 با ضرایب خاص که بر روی دو فریم متوالی از یک سری تصاویر ترتیبی عمل می‌کنند قابل محاسبه‌اند [۹].

روش محاسبه شار نوری هرن و شانک با استفاده از

FPGA

الگوریتم‌های محاسبه شار نوری معمولاً شامل فرایندهای تکراری و مستقل از هم می‌باشند به‌نحوی که محاسبه بخش‌های مختلف

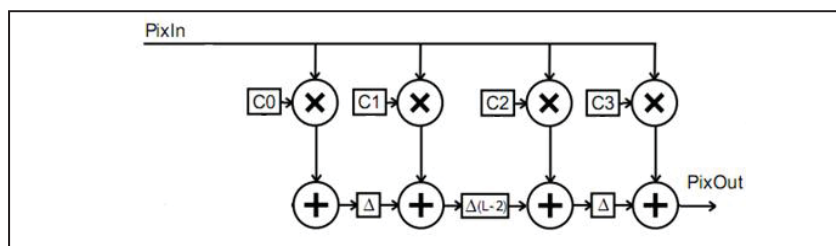


شکل ۴- ساختار موازی الگوریتم هرن و شانک برای پیاده‌سازی بر روی FPGA

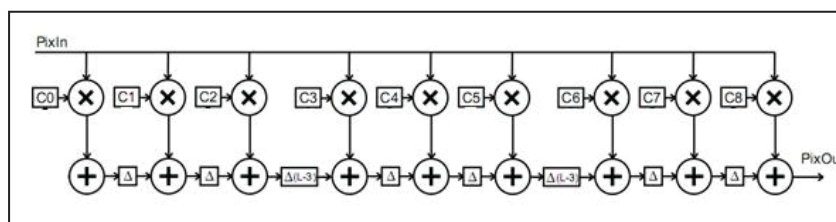
پیاده‌سازی فیلتر 3×3 نشان داده شده است. شکل (۳) بر روی ماتریس‌های u و v است. برای این منظور نیز می‌توان از روش Pipeline استفاده کرد؛ با این تفاوت که در این حالت به ۲ تأخیردهنده خط و ۵ تأخیردهنده پیکسل نیاز است. در حالت کلی می‌توان از طرح ارائه شده در شکل (۶) برای محاسبه لاپلاسیان در FPGA استفاده کرد. ساختار فیلتر نشان داده شده در شکل (۶) در واقع تعمیمی از فیلتر 2×2 نشان داده شده در شکل (۵) می‌باشد. استفاده از روش Pipeline علاوه بر بهینه کردن موردنظر، باعث افزایش چشم‌گیر سرعت محاسبات می‌شود. در این روش با اعمال هر پیکسل به سیستم Pipeline یک خروجی نیز ایجاد خواهد شد که می‌تواند مربوط به پیکسل فعلی یا پیکسل‌های اعمال شده قبلی باشد.

محاسبات نهایی برای محاسبه u و v

پس از محاسبه لاپلاسیان u و v و مشتق‌های تصویر با استفاده از فیلترهای Pipeline طراحی شده در FPGA، می‌توان از خروجی این فیلترها و روابط ۲ و ۳ برای محاسبه شار نوری نهایی استفاده کرد. باید توجه داشت که در این روش باید همزمانی لازم نیز بین خروجی فیلتر 2×2 و خروجی فیلتر 3×3 وجود داشته باشد. همان‌گونه که شکل (۷) نشان می‌دهد خروجی فیلتر 2×2 برای مؤلفه (i, j) زمانی به دست می‌آید که عنصر (i, j) از تصویر به FPGA اعمال شود و خروجی فیلتر 3×3 برای مؤلفه (i, j) ماتریس‌های u و v زمانی محاسبه می‌شوند که مؤلفه $(i+1, j+1)$ برای انجام محاسبات به بلوک فیلتر 3×3 اعمال شود.



شکل ۵- روند محاسبه فیلتر 2×2 در حالت Pipeline



شکل ۶- روند کلی محاسبه فیلتر 3×3 تصویر با استفاده از روش Pipeline

پیاده‌سازی مشتق‌های تصویر با استفاده از FPGA

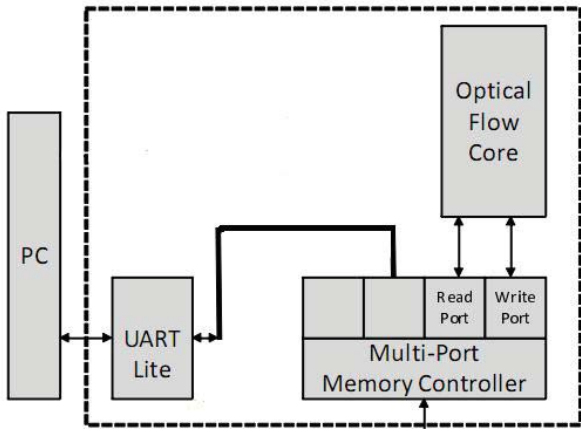
مشتق‌های تصویر در روش هرن و شانک، جهت محاسبه شار نوری، در واقع اعمال یک فیلتر 2×2 بر روی تصویر می‌باشد، روشی که در این مقاله برای اعمال این فیلتر بر روی تصویر پیشنهاد می‌شود، یک روش Pipeline است. در شکل (۵) با ورود هر پیکسل تمامی حاصل‌ضرب‌های مربوط به فیلتر برای آن محاسبه می‌شوند؛ سپس مقادیر محاسبه شده وارد تأخیردهنده‌هایی می‌شوند تا بتوانند در لحظه مناسب در محاسبات وارد شوند. از آنجا که فیلتر مربوط به مشتق‌های تصویر 2×2 است، لذا به یک تأخیردهنده خط و دو تأخیردهنده پیکسل نیاز داریم، تأخیردهنده خط حاصل‌ضرب به دست آمده به ازای هر پیکسل را، تقریباً پس از طی شدن یک ردیف کامل از تصویر در محاسبات وارد می‌کند. تأخیردهنده خط در واقع ایجادکننده تأخیر برای تعدادی پالس ساعت (معادل یک سطر از تصویر) می‌باشد. برای پیاده‌سازی چنین تأخیری در FPGA از حافظه‌های FIFO^۱ با عمقی معادل طول تصویر استفاده شد. تأخیردهنده‌های پیکسل نیز زمان اعمال حاصل‌ضرب مذکور را با یک واحد تأخیر به عقب می‌ندازد. روند کلی محاسبه یک فیلتر 2×2 در شکل (۵) به تصویر کشیده شده است. که در آن ثوابت C_0, C_1, C_2, C_3 ضرایب فیلتر 2×2 می‌باشند. L بیانگر طول تصویر و Δ بیانگر تأخیر است. $\Delta(L-2)$ بیانگر تأخیر به اندازه طول تصویر منهای ۲ و Δ بیانگر تأخیر به اندازه یک پیکسل است.

محاسبه میانگین سرعت‌های u و v در FPGA

میانگین مؤلفه‌های سرعت تصویر در روش هرن و شانک نیز در واقع

1- First In First Out

الگوریتم مورد نظر در نهایت بر روی تراشه XC3SD3400A برنامه‌ریزی و نتایج مربوطه مورد بررسی قرار گرفتند. روند پیاده‌سازی شار نوری در این مقاله مطابق شکل (۸) می‌باشد.



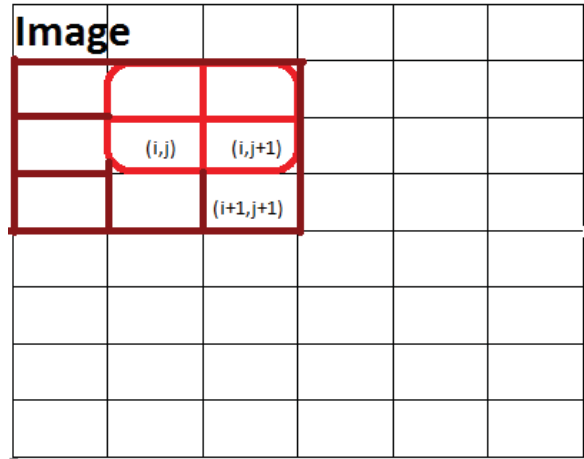
شکل ۸- روند پیاده‌سازی الگوریتم شار نوری بر روی FPGA

پس از ذخیره‌سازی تصاویر متوالی و ماتریس‌های u و v اولیه در فضای RAM موجود در تراشه XC3SD3400A، شار نوری طبق روند ذکر شده در بخش‌های قبل محاسبه می‌شود.

همان‌گونه که شکل (۸) نشان می‌دهد در این پیاده‌سازی از حافظه‌های فقط خواندنی (ROM) برای ذخیره‌سازی تصاویر، و از حافظه‌هایی با قابلیت خواندن و نوشتن همزمان (Dual Port RAM) برای ذخیره‌سازی ماتریس‌های u و v استفاده شده است.

ارتباط با پورت سریال و انتقال u و v نهایی به رایانه و ارزیابی سخت‌افزاری الگوریتم

جهت ارزیابی نهایی و اطمینان از صحت خروجی، با استفاده از زبان توصیف رفتاری VHDL برنامه رابط سریال جهت ارتباط با رایانه تهیه شد و نتایج الگوریتم از طریق رابط سریال (UART) به رایانه انتقال داده شدند تا از صحت سخت‌افزار ایجاد شده جهت محاسبه شار نوری هرن و شانک اطمینان حاصل شود. شار منتقل شده بر روی پورت سریال رایانه توسط رابط^۱ ایجاد شده به وسیله نرم‌افزار متلب^۲، برای انجام مقایسه لازم، توسط نرم‌افزار دریافت و مورد بررسی قرار گرفت.



شکل ۷- تصویر و نحوه اعمال فیلترهای 2×2 و 3×3 بر روی آن

باید توجه داشت که همزمانی مذکور حتماً ایجاد شود. این همزمانی با اعمال یک خط تأخیر بر روی مشتق‌های تصویر قابل دستیابی است. این تأخیردهنده با اعمال تأخیر لازم بر روی مقادیر محاسبه شده برای مشتق‌های تصویر این مقادیر، زمان مناسبی که لاپلاسیان u و v مربوط به آن‌ها نیز حساب شد مورد استفاده قرار می‌گیرند. پس از همزمانی‌های مذکور، محاسبات ریاضیاتی نهایی باید انجام گیرد تا با استفاده از مشتق‌های تصویر و لاپلاسیان به دست آمده از تصویر بتوان u و v مربوط به هر پیکسل را محاسبه کرد. از آنجا که روش هرن و شانک یک روش بسیار حساس برای به دست آوردن شار نوری است، لذا محاسبات باید با دقت اعشاری بالا انجام شود. در این مقاله برای دستیابی به دقت بالا و انجام محاسبات ریاضیاتی از جمله ضرب و تقسیم در FPGA از بلوک‌های ضرب‌کننده و تقسیم‌کننده موجود در تراشه XC3SD3400A استفاده شده است. این بلوک‌ها توسط شرکت ارائه‌دهنده تراشه به صورت بهینه در تراشه مذکور پیاده‌سازی شده‌اند و قابل استفاده می‌باشند. استفاده از این بلوک‌ها موجب بهینه‌سازی طرح از نظر فضای اشغال شده در تراشه و سرعت محاسبات خواهد شد.

شبیه‌سازی و پیاده‌سازی الگوریتم محاسبه شار نوری

روند ارائه شده در بخش‌های قبل، نحوه پیاده‌سازی شار نوری را با کمترین تأخیر ممکن می‌سازد. در این مقاله، برای شبیه‌سازی و پیاده‌سازی روند ارائه شده، ابزار ISE موجود برای محصولات شرکت Xilinx و زبان توصیف سخت‌افزار VHDL مورد استفاده قرار گرفتند. ابزار ISim نیز پس از تهیه کد VHDL مورد نیاز جهت شبیه‌سازی طرح کلی مورد استفاده قرار گرفت. ابزار ISim یک ابزار مناسب برای شبیه‌سازی طرح‌های مبتنی بر FPGA است که توسط شرکت Xilinx در اختیار کاربران قرار گرفته است.

1- Interface
2- MatLab

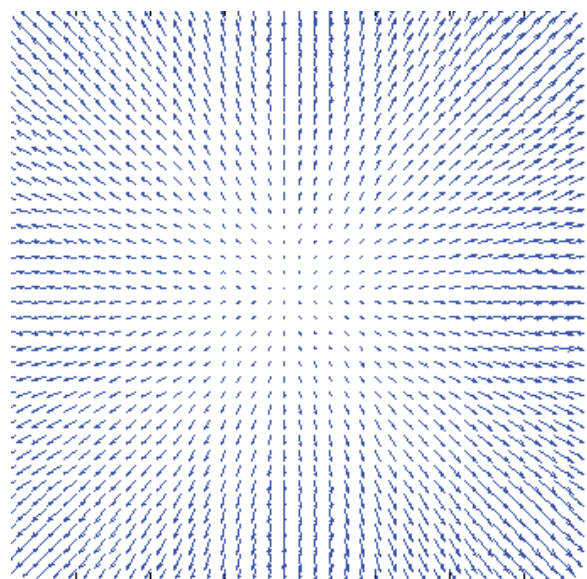
مقایسه شار صحیح تصاویر استاندارد با نتایج حاصل از

پیاده‌سازی شار نوری هنر و شانک

بارن و همکارانش تصاویر استاندارد را به همراه شار صحیح آن تصاویر برای ارزیابی الگوریتم‌های مختلف محاسبه شار نوری ارائه کرده‌اند [۶]. ما نیز در این مقاله از تصاویر استاندارد برای ارزیابی الگوریتم‌منتخب استفاده می‌کنیم. در این مقاله از تصاویر استاندارد درخت، در حالت جابه‌جایی صحنه به سمت راست (translating tree) و حالت همگرا شدن دوربین نسبت به صحنه (diverging tree)، استفاده شده است. شکل‌های (۹) تا (۱۱) تصویر استاندارد درخت را به همراه شار صحیح برای هر دو حالت مذکور نشان می‌دهند.



شکل ۹- تصویر استاندارد مورد ارزیابی



شکل ۱۰- شار صحیح برای diverging tree

شکل ۱۱- شار صحیح برای translating tree

شکل (۱۲) شار حاصل از پیاده‌سازی FPGA را برای تصاویر استاندارد درخت نمایش داده است. باید توجه داشت که شار نوری بارن و همکارانش در واقع یک شار صحیح مورد انتظار برای فریم‌های استاندارد ارائه شده است. شار نوری بارن خروجی الگوریتم خاصی نمی‌باشد، بلکه شار صحیحی است که می‌توان الگوریتم‌های مختلف را توسط آن ارزیابی کرد.

رودل و همکارانش^۱ معادله‌ای را برای به‌دست آوردن میزان خطای الگوریتم‌های محاسبه شار نوری ارائه کردند [۱۱]. رابطه (۴) بیانگر معادله مذکور است.

$$E = \sqrt{\frac{1}{HW} \sum ((Uc - Ue)^2 + (Vc - Ve))} \quad (4)$$

در این رابطه، H و W به ترتیب بیانگر طول و عرض تصویر و Uc و Ue و Vc شار صحیح برای هر پیکسل و Ve و Ue شار تخمین‌زده توسط الگوریتم مورد نظر می‌باشد. E نیز مقدار خطای الگوریتم (RMS) بر حسب پیکسل می‌باشد. جدول (۱) نتایج حاصل از پیاده‌سازی را برای الگوریتم منتخب نشان می‌دهد.

جدول ۱- نتایج حاصل از پیاده‌سازی الگوریتم محاسبه شار هنر - شانک برای مقایسه دقت پیاده‌سازی

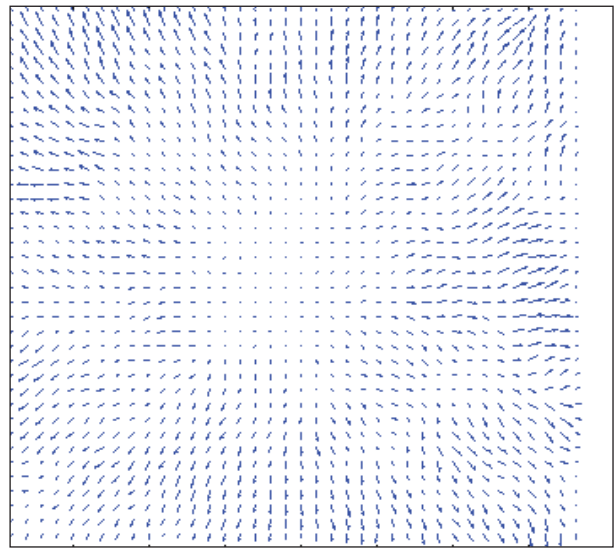
تصویر	RMS
Diverging tree	۰,۸۲۷۰
Translating tree	۲,۳۲۵۶

حرکت از نوع انتقالی نمی‌باشد و خطا در این حالت زیاده‌تر از حالتی است که در آن جسم یا دوربین حرکت انتقالی داشته باشند. در مجموع، وجود تفاوت بین شار صحیح بارن و شار نوری هرن و شانک را می‌توان حساسیت زیاد شار نوری هرن و شانک به نویز موجود در تصویر و وجود حرکت‌های بزرگ در تصویر دانست.

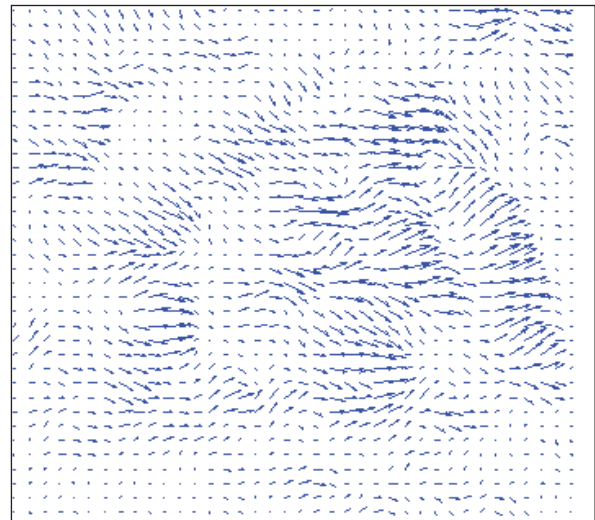
مقایسه زمان اجرای الگوریتم هرن و شانک در نرم‌افزار

متلب و FPGA

زمان محاسبه شار برای دو فریم با ابعاد 128×128 در نرم‌افزار متلب موجود در سیستم عامل ویندوز نصب شده بر روی سیستم Pentium(R) Dual Core CPU، 0.11046 ثانیه است. این در حالی است که استفاده از روش pipeline می‌تواند بر حسب سرعت حافظه مورد استفاده برای ذخیره‌سازی تصویر و ماتریس‌های u و v ، FPGA مورد استفاده جهت پردازش و دوربین به کار رفته به صورت بلادرنگ و یا همراه با تأخیر انجام پذیرد. در این مقاله از حافظه‌های داخلی موجود در تراشه XC3SD3400A برای ذخیره‌سازی تصویر و ماتریس‌های u و v استفاده شد. امکان دسترسی به داده موجود در بلوک‌های حافظه داخلی تراشه، تنها با یک پالس ساعت امکان‌پذیر است که این مسئله موجب حذف تأخیر ناشی از فراخوانی داده موجود در حافظه و نیز باعث بهینه شدن طرح سخت‌افزاری از نظر تأخیر اجرا خواهد شد. تأخیر ناشی از پیاده‌سازی سخت‌افزاری شار نوری هرن و شانک در فرکانس کاری ۳۳ مگاهرتز (فرکانس کاری دوربین موجود بر روی برد) در این مقاله به دلیل استفاده از ساختار Pipeline تقریباً صفر است. این موضوع برتری پیاده‌سازی سخت‌افزاری شار نوری هرن و شانک را نسبت به پیاده‌سازی نرم‌افزاری نشان می‌دهد. در تحقیق حاضر، تصاویر اخذ شده توسط دوربین به نحوی توسط FPGA نمونه‌برداری شدند که بتوان در نهایت به تصویری با سایز 128×128 در سخت‌افزار دست پیدا کرد. تصاویر به صورت سیاه و سفید و با دقت ۸ بیت از دوربین دریافت شده و شار مربوط به آن‌ها محاسبه می‌شود. در این تحقیق، از حافظه داخلی^۲ تعبیه شده درون FPGA جهت ذخیره‌سازی تصویر و ماتریس‌های u و v استفاده شده است. در این حالت پروژه محدود به استفاده از ۲۸۳ کیلوبایت حافظه داخلی است، از سویی دسترسی به داده موجود در حافظه داخلی FPGA تنها با یک پالس ساعت امکان‌پذیر است که این امر موجب کمینه شدن تأخیر موجود در سیستم محاسبه شار نوری می‌شود. شکل (۱۳) تصویر برد آزمایشگاهی مورد استفاده جهت پیاده‌سازی شار نوری هرن و شانک را به تصویر کشیده است. این برد آزمایشگاهی مجهز به دوربین، رابط‌های پرسرعت برای ارتباط با رایانه، تراشه XC3SD3400A و... می‌باشد.



(الف)



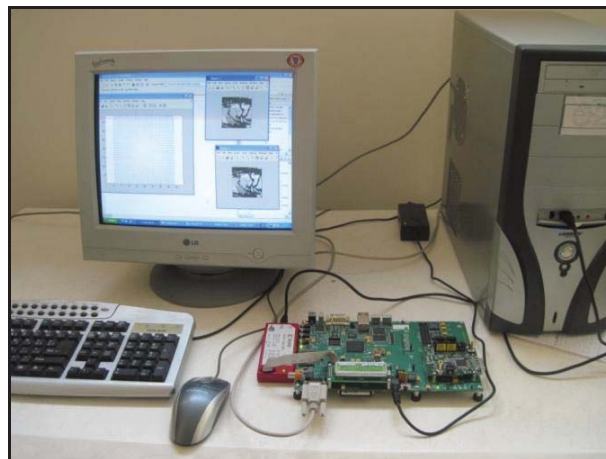
(ب)

شکل ۱۲- (الف) شار حاصل از پیاده‌سازی حاصل برای **diverging tree** و (ب) برای **translating tree**

مؤلفه‌های شار نوری در صورتی بیانگر حرکت صحیح در تصویر خواهند بود که حرکات موجود در تصاویر متغیر با زمان، مربوط به سطوح لامبرتین^۱ با حرکاتی کاملاً انتقالی و تحت روشنایی زمان - مکانی یکنواخت باشند [۱۲]. نتایج جدول (۱) نیز نشان می‌دهد که میانگین خطا برای کل پیکسل‌های یک تصویر در حالتی که جسم موجود در صحنه حرکتی از نوع انتقالی (Translating motion) داشته باشد، کمتر از حالتی است که جسم به دوربین نزدیک شود. چرا که در حالت همگرا شدن دوربین به صحنه (Diverging)، ماهیت

مراجع

1. C. Fermuller, D. Shulman, Y. Aloimonos, The statistics of optical flow, *Comput. Vis. ImageUnderstand.* 82 (5) (2001) 1-32.
2. M. Fleury, A. Clark, A. Downton, Evaluating optical-flow algorithms on a parallel machine, *ImageVis. Comput.* 19 (3) (2001) 131-143.
3. B. McCane, K. Novins, D. Crannitch, B. Galvin, On benchmarking optical flow, *Comput. Vis. Image Understand.* 84 (1) (2003) 126-143.
4. D. J. Fleet and K. Langley. "Recursive filters for Optical Flow". *IEEE Transactions PAMI*, 17(1):61-67, (1995).
5. Hongche Liu, Tsai-Hong Hong, Martin Herman, Ted Camus, and Rama Chellappa. "Accuracy vs efficiency trade-offs in optical flow algorithms". *Computer Vision: CVIU*, 72(3):271-286, (1998).
6. J. L. Barron, D. J. Fleet, and S. S. Beauchemin. "Performance of Optical Flow Techniques - Systems and Experiment". *International Journal of Computer Vision*, 12(1):4377, (1994).
7. Michael Chisholm. "Calculating Optical Flow using FPGA", ANU College of Engineering and Computer Science, COMP3750, Project Work in Computer Systems, (2008).
8. Rafael C. Gonzalez, Richard E. Woods. "Digital Image Processing", 2nd. ed., 161-162.
9. Berthold K.P.Horn and Brian G.Schunck. "Determining Optical Flow". Massachusetts Institute of Technology, Cambridge, MA 02130, U.S.A, 185-203
10. <http://www.xilinx.com>
11. N. Roudel, F. Berry, J. Serot, "Hardware Implementation of a Real Time Lucas and Kanade Optical Flow".
12. A. Verri, T. Poggio, "Motion Field and Optical Flow Qualitative Properties", *IEEE Trans. Pattern Anal. Mach. Intell.* 11 (5) (1989) 490-498.



شکل ۱۳-۱ تصویر برد آزمایشگاهی مورد استفاده به همراه دوربین به کار رفته

نتیجه گیری

در این مقاله، روشی بهینه برای پیاده‌سازی شار نوری هرن و شانک مطرح شد. از آنجایی که شار نوری پیش‌نیازی برای بسیاری از سیستم‌های نظامی از جمله سیستم‌های ناوبری خودکار، رهگیری هدف، سیستم‌های امنیتی و... می‌باشد، سرعت محاسبه شار از مؤلفه‌های اساسی این‌گونه سیستم‌هاست. لذا پیاده‌سازی الگوریتم‌های محاسبه شار نوری بر روی سخت‌افزاری سریع از اهمیت ویژه‌ای برخوردار است. در این مقاله، پس از پیاده‌سازی شار بر روی FPGA، نتایج حاصل از پیاده‌سازی سخت‌افزاری با شار صحیح مورد انتظار برای تصاویر بارن مورد بررسی قرار گرفتند. پیاده‌سازی سخت‌افزاری نتایج مناسبی در پی داشت که این امر امکان استفاده از این سیستم را در کاربرهای دقیق و بلادرنگ فراهم می‌سازد.

Horn – Schunk Optical Flow Implementation on FPGA

A. Bahrami¹

M. R. Mahzoun²

A. R. Amin²

M. Rahimi³

J. Arab³

Abstract

Calculating the speed of all points of moving objects is one of the most important subjects in image processing fields. Some methods called sequential images optical flow extraction are declared related to this subject. Optical flow has many applications in military science such as: object tracking, PIV, odometry or range finding, UAV monitoring, robot control and Due to the passive nature of image processing systems and image flow calculations, using such systems in the passive defense applications will be appropriate. Although many successful methods of optical flow have been introduced so far, the implementation of these methods, due to their calculation complexity and their slow response of the system to its operator, is not usually economical.. In this paper , Horn and Schunk algorithm is selected for optical flow implementation on FPGA as a very suitable algorithm. Selected FPGA is from Spartan 3A DSP3400 family suitable for image processing tasks. After implementation on FPGA, results will be evaluated using standard images and their correct optical flow

Key Words: *optical Flow, FPGA, Velocity Determination and Image Processing*

1- MS in Electronics, Imam Hossein Comprehensive University (Pbh), Writer in Charge, Tehran, Iran (Email: Bahrami.Amin@yahoo.com).

2- Assistant Professor and Academic Member of the Faculty and Research Center of ICT, Imam Hossein Comprehensive University (Pbh).

3- MS in Electronics, Imam Hossein Comprehensive University (Pbh).