Electronical & Cyber Defence
Vol. 12, No. 4, Winter 2025, Serial No. 48

Improved implementation of image processing algorithm with HLS software for use in optical seeker

M. Yousefvand [1], M.Nasiri[2]* , R.Bibak[3]

*Assistant Professor, Imam Hussein , University, Tehran, Iran

Received: 2024/08/22, Revised: 2025/12/10, Accepted: 2025/01/12, Published: 2025/02/01)

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.4.2.6

ABSTRACT

Current image processing systems have the ability to fast process images with high imaging rates. To reduce calculation time and increase processing speed, algorithms can be implemented on hardware accelerators such as FPGA. The hardware implementation of image processing algorithms should be done optimally with the aim of increasing the processing speed, reducing the resources consumed and, as a result, reducing the cost of the used processor. In the application of target detection, the optical seeker will be able to detect, recognize and track the target by using image processing algorithms and comparing the current image information of the camera with the information of the desired target image that has already been stored in its processor memory. In this process, SIFT is an algorithm for matching applications, of which the DOG section is one of its subsections. DOG alone covers more than 80% of the execution time of the SIFT algorithm. The reason for this is that many Gaussian filters are multiplied in the input image. The proposed architecture is presented in such a way that only two DSP48s are used for the RTL implementation of each 15x15 Gaussian filter. The advantage of reducing the number of resources related to the cheapness of FPGA has been used. HLS tool has been used to implement the proposed architecture.

Keywords: Seeker, SIFT Algorithm, DOG, HLS, DSP48

*Corresponding Author Email: mnasiri@ihu.ac.ir

"پدافند الکترونیکی و سایبری"

سال دوازدهم، شماره ۴، زمستان ۱۴۰۳، ص۱۹-۱۳

شاپا الکترونیکی: ۸۹۷۹-۲۹۸۰ شاپا چاپی: ۴۳۴۷-۲۳۲۲

علمی –پژوهشی

پیاده‌سازی بهبودیافته الگوریتم پردازش تصویر با نرم‌افزار HLS جهت استفاده در جستجوگر اپتیکی

مهدی یوسف‌وند[۱]، مهدی نصیری[۲*]، روح‌الله بی‌باک[۳]

۱- دانشجوی کارشناسی‌ارشد،۲- استادیار دانشگاه جامع امام حسین (ع)،تهران،ایران ۳- کارشناسی ارشد، دانشگاه تربیت مدرس، تهران، ایران

(دریافت: ۱۴۰۳/۰۶/۰۱، بازنگری: ۱۴۰۳/۰۹/۲۰، پذیرش: ۱۴۰۳/۱۰/۲۳، انتشار: ۱۴۰۳/۱۱/۱۳)

DOR: https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.4.2.6 .۱

چکیده

سامانه‌های پردازش تصویر کنونی بایستی توانایی پردازش سریع تصاویر با نرخ تصویربرداری بالا را داشته باشند. برای کاهش زمان محاسبات و افزایش سرعت پردازش می‌توان الگوریتم‌ها را بر روی شتاب‌دهنده‌های سخت‌افزاری مانند FPGA پیاده‌سازی نمود. پیاده‌سازی سخت‌افزاری الگوریتم‌های پردازش تصویر بایستی به‌صورت بهینه و باهدف افزایش سرعت پردازش، کاهش منابع مصرفی و درنتیجه کاهش هزینه پردازنده به‌کاررفته صورت گیرد. در کاربرد آشکارسازی هدف، جستجوگر اپتیکی با استفاده از الگوریتم‌های پردازش تصویر و مقایسه اطلاعات تصویر فعلی دوربین با اطلاعات موجود از تصویر هدف موردنظر که قبلاً در حافظه پردازشگر آن ذخیره شده است قادر به آشکارسازی، تشخیص و ردگیری هدف خواهد بود. در این فرایند، SIFT الگوریتمی برای کاربردهای تناظریابی است که بخش DoG یکی از زیر بخش‌های آن است. DoG به‌تنهایی بیش از 80% زمان اجرای الگوریتم SIFT را در بر می‌گیرد. دلیل این امر این است که فیلترهای گاوسین به تعداد زیادی در تصویر ورودی ضرب می‌شود. معماری پیشنهادی به‌گونه‌ای است که برای پیاده‌سازی RTL هر فیلتر گاوسین ۱۵*۱۵ تنها از دو واحد DSP48 استفاده می‌شود. مزیت کاهش تعداد منابع مربوط به ارزان شدن FPGA به کار گرفته شده است. برای پیاده‌سازی معماری پیشنهادی ابزار HLS به‌کاررفته است.

کلیدواژه‌ها: جستجوگر، الگوریتم SIFT، DoG، HLS، DSP48

۱- مقدمه

باتوجه‌به اینکه هدایت موشک در فاز میانی به‌وسیله دستگاه‌هایی نظیر سیستم ناوبری اینرسی صورت می‌پذیرد، خطای روانه‌سازی و هدایت موشک در حین پرواز زیاد شده و عملاً امکان اصابت دقیق به هدف موردنظر وجود ندارد. برای اینکه اهداف ثابت زمینی به‌وسیله موشک بادقت و با حداقل خطای ممکن مورد اصابت قرار گیرد بایستی از حسگرهای کمکی استفاده شود [۱]. به‌منظور رسیدن به این هدف لازم است که هدایت موشک به سمت هدف در فاز نهایی بر عهده جستجوگر قرار گیرد. وظیفه جستجوگر تعیین مکان هدف و آشکارسازی و ردگیری آن تا لحظه اصابت است. جستجوگر، سیگنال‌های خطا را برای سیستم هدایت موشک تولید می‌کند. جستجوگرهای فعال رادیویی، انرژی لازم را برای آشکارسازی و ردگیری هدف باتوجه‌به برد مورد نیاز، ارسال کرده و باتوجه‌به انرژی برگشتی از هدف، آن را آشکارسازی و ردگیری می‌نمایند. با استفاده از این جستجوگرها نمی‌توان اهداف ثابت زمینی را مورد اصابت قرارداد زیرا توان کلاتر زمین بسیار بیشتر از توان برگشتی از هدف است. بنابراین هدف در کلاتر زمین گم شده و به‌وسیله این جستجوگر آشکار نمی‌شود [۲]. جستجوگرهای غیرفعال رادیویی علیه منابع تشعشع رادیویی که در فرکانس کاری جستجوگر کار می‌کنند، مورداستفاده قرار می‌گیرند. معمولاً انواع رادارها جزء اهداف این

* رایانامه نویسنده مسئول: mnasiri@ihu.ac.ir

نوع جستجوگرها هستند. باتوجه‌به قابلیت‌های رادارهای هوشمند امروزی به‌محض شلیک موشک ضد تشعشع[1] به سمت آن‌ها سعی در مقابله و جمینگ جستجوگر موشک می‌نمایند. یکی از ساده-ترین روش‌هایی که رادار طعمه برای مقابله با این موشک‌ها به کار می‌گیرد، خاموش‌کردن فرستنده خود است. جستجوگر غیر-فعال در صورت خاموش‌شدن فرستنده رادار طعمه و قطع سیگنال ارسالی، کارایی خود را ازدست‌داده و عملاً نمی‌تواند هدف را مورد اصابت قرار دهد [۳]. جستجوگرهای نیمه‌فعال رادیویی در سامانه‌های پدافندی به‌منظور درگیری با اهداف هوایی مورداستفاده قرار می‌گیرند، بنابراین قابلیت درگیری با اهداف ثابت زمینی را ندارند. باتوجه‌به نقاط ضعف جستجوگرهای رادیویی که به آن‌ها اشاره شد نمی‌توان از آن‌ها جهت انهدام اهداف ثابت زمینی استفاده کرد و بایستی از یکی دیگر از انواع جستجوگر استفاده نماییم. جستجوگرهای اپتیکی یکی از انواع جستجوگرها هستند که می‌توانند به این منظور مورداستفاده قرار گیرند. حال به بررسی عملکرد و قابلیت‌های جستجوگرهای اپتیکی می‌پردازیم. در علم مهندسی فوتونیک، به طول‌موج حدود 400 nm تا 700 nm ناحیه مرئی[2] و به طول‌موج حدود 700 nm تا 1 mm ناحیه مادون‌قرمز[3] می‌گویند. لازم به ذکر است که جستجوگرهای لیزری نیز جزء این دسته از جستجوگرها هستند که به‌صورت نیمه‌فعال کار می‌کنند. برای اینکه جستجوگر لیزری بتواند هدف ثابت زمینی را مورد اصابت قرار دهد بایستی هدف موردنظر توسط یک فرستنده لیزری روشن شود و جستجوگر با دریافت امواج لیزری منعکس‌شده از هدف، آن را آشکارسازی و ردگیری نماید. تاکنون استفاده از جستجوگرهای لیزری در هیچ‌کدام از سامانه‌های موشکی زمین‌به‌زمین دنیا برای هدف قراردادن اهداف ثابت زمینی استفاده نشده است و این جستجوگرها معمولاً در بمب‌ها و موشک‌های هوا به زمین که فرستنده (روشن‌کننده هدف) آن‌ها بر روی هواپیما، بالگرد و یا پهپاد قرار دارد مورداستفاده قرار می‌گیرند. یکی از معایب عمده این جستجوگرها این است که بایستی روشن‌کننده که شامل هواپیما، بالگرد و یا پهپاد است تا لحظه اصابت در منطقه عملیات حضور داشته باشد و حضور آن ریسک عملیات را بالا می‌برد. جستجوگرهای اپتیکی مرئی و مادون‌قرمز به‌صورت غیرفعال، هدف را آشکارسازی و ردگیری می‌کنند. معمولاً به‌منظور درگیری با اهداف هوایی نظیر انواع موشک، هواپیما و بالگرد از جستجوگرهای مادون‌قرمز استفاده می‌شود. دلیل این کار، وجود منبع گرمایی ناشی از احتراق موتور این اهداف است که باعث می‌شود جستجوگر بتواند به‌سهولت و بادقت بالا این اهداف را آشکارسازی و ردگیری نماید[۴]. باتوجه‌به مطالب گفته‌شده، جستجوگر اپتیکی باقابلیت عملکرد در ناحیه مرئی طیف امواج الکترومغناطیسی، بهترین انتخاب برای هدف قراردادن اهداف ثابت زمینی است. این جستجوگر اپتیکی با استفاده از الگوریتم-های پردازش تصویر و مقایسه اطلاعات تصویر فعلی دوربین با اطلاعات موجود از تصویر هدف موردنظر که قبلاً در حافظه پردازشگر آن برنامه‌ریزی شده است قادر به آشکارسازی، تشخیص و ردگیری هدف خواهد بود. لازمه این عملیات، توصیف تصویر هدف است. پژوهش حاضر الگوریتم SIFT[4] را که الگوریتمی جهت تعیین و توصیف دقیق تصویر هدف است، مورد بررسی قرار می‌دهد.

۲- روش تحقیق

الگوریتم SIFT یک الگوریتم دومرحله‌ای است که ابتدا ویژگی‌های تصویر را استخراج می‌کند و سپس هرکدام از آن‌ها را توصیف می‌کند. در بخش مربوط به تعیین ویژگی‌های تصویر توسط الگوریتم SIFT اولین قدم تعیین مکان‌هایی است که تغییرات محلی مهم در جلوه‌های بصری آن‌ها وجود دارد. این مکان‌ها بهترین کاندید برای نقاط کلیدی SIFT هستند. برای به‌دست‌آوردن این نقاط کلیدی بایستی یک هرم تصویر DoG[5] بسازیم که گرادیان میدان تصویر را تقریب می‌زند [۵،۶].

به طور مثال اگر یک تصویر مانند $I(x,y)$ موجود باشد در قدم اول باید فضای مقیاس تشکیل داده شود. به این منظور باید تصاویری با مقیاس‌های مختلف از روی تصویر اصلی ساخته شود که تعداد این تصاویر مقیاس یکی از پارامترهایی است که با توجه به‌اندازه تصویر اصلی تعیین می‌شود. به طور مثال اگر تعداد تصاویر مقیاس برابر ۴ باشد در این صورت، هر بار تصویر نصف می‌شود و آنگاه خود تصویر $I(x,y)$ و مقیاس‌های $I/2$ و $I/4$ و $I/8$ فضای مقیاس را تشکیل خواهند داد [۷،۸].

حال باید تصویر ورودی $I(x,y)$ با یک کرنل گاوسین $K(x,y;\delta)$ کانال و شود که در آن δ انحراف معیار کرنل گاوسین است. اگر نتیجه تصویر به طور گاوسین فیلتر شده را با $G(x,y;\delta)$ نشان دهیم آنگاه به رابطه (۱) خواهیم رسید:

$$(x,y;\delta) = \text{conv2}(I(x.y) ,K(x,y;\delta))$$

رابطه (۱)

$$K(x,y;\delta) = \frac{1}{2\pi\delta^2}\, e^{-(x^2+y^2)/2\delta^2}$$

[1] Anti Radiation Missile(ARM)
[2] Visible
[3] Infrared (IR)

[4] Scale Invariant Feature Transform
[4] Difference-of-Gaussian

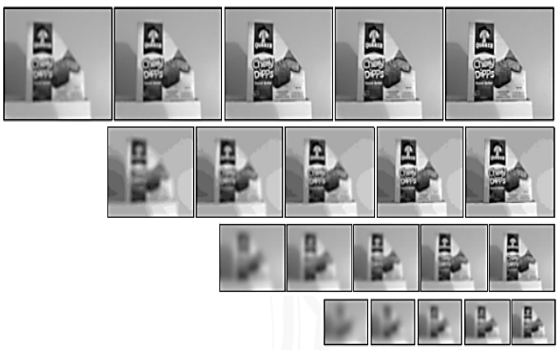
که در آن $conv2(,)$ نشان‌دهنده کانولوشن دوبعدی است.

مقدار انحراف معیار برای اعمال به تصاویر در فضای مقیاس به‌صورت دلخواه انتخاب نمی‌شود و به این صورت است که از هر اکتاو به اکتاو بعدی مقدار انحراف معیار دوبرابر می‌شود و مقدار انحراف معیار هر فیلتر از رابطه (۲) به دست می‌آید:

$$\delta_{octave} = 2^{(m-1)}\,\delta_0$$
$$\delta = k^{(n-2)}\,\delta_{octave} \qquad \text{رابطه (۲)}$$
$$k = 2^{(1/s)}$$

در این روابط δ_0 انحراف معیار اولیه است و m شماره اکتاو است. به‌عنوان‌مثال در اکتاو اول که مقیاس همان اندازه تصویر اصلی است انحراف معیار اکتاو برابر انحراف معیار اولیه است. پارامتر n شماره فیلتری است که انحراف معیار آن را در اکتاو موردنظر می‌خواهیم محاسبه نماییم که از یک شروع می‌شود و s نیز تعداد فیلترهای هر اکتاو است.

تعداد انحراف معیارهای بکار رفته یکی دیگر از پارامترهایی است که بنا به‌اندازه تصویر تعیین می‌گردد. در این مثال اگر فرض شود که ۵ انحراف معیار مختلف موجود باشد و یا به‌عبارت‌دیگر به هر تصویر مقیاس یافته ۵ فیلتر گاوسین با انحراف معیارهای مختلف اعمال شود آنگاه مانند شکل (۱)، ۴ دسته تصویر با مقیاس‌های متفاوت وجود خواهد داشت که هرکدام دارای ۵ تصویر هستند که به هرکدام از این دسته تصاویر ۵تایی یک اکتاو گفته می‌شود. لازم به ذکر است که مقدار انحراف معیار فیلتر اعمالی در همواری یا تاری تصویر تأثیر به سزایی دارد به‌نحوی‌که هر چه مقدار انحراف معیار بزرگ‌تر باشد تصویر فیلتر شده دارای تاری بیشتری خواهد بود.

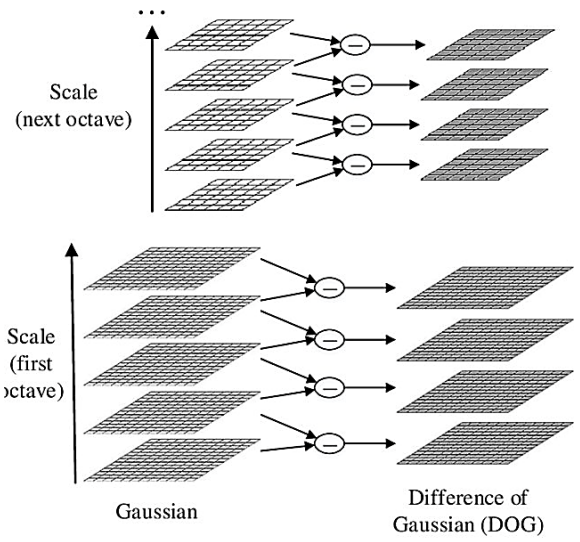


شکل (۱). تصویر فیلتر شده ۴ اکتاو (هر اکتاو شامل ۵ فیلتر)

همان‌طوری که در شکل (۲) دیده می‌شود پس از تشکیل فضای مقیاس، هر دو تصویر مجاور در داخل یک اکتاو تفریق می‌شود که این عملیات تفاضل گاوسین (DoG) را که فضای مقیاس جدید است تشکیل خواهد داد. به‌صورت ریاضی هر یک از تصاویر این فضای مقیاس جدید از رابطه (۳) به دست می‌آید:

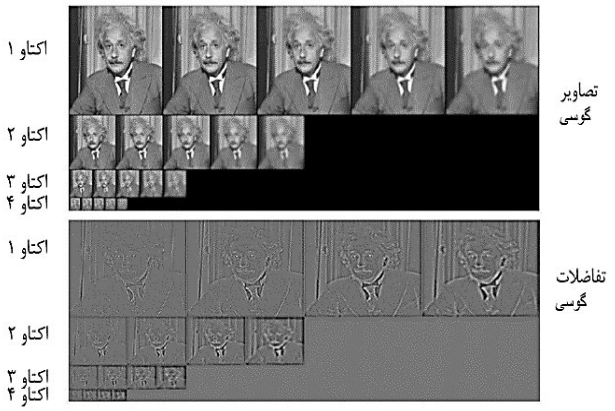
$$(x,y;\delta) = (x,k;\delta) - G(x,y;\delta) \qquad \text{رابطه (۳)}$$

که در آن k یک عامل ضربی ثابت است [۹].



شکل (۲). فضای مقیاس و به دست آوردن عملیات تفاضل گاوسین از این فضا

اعمال فیلتر تفاضل گاوسین برای پیداکردن نقاط اکسترمم در فضای مقیاس صورت می‌پذیرد. همان‌طوری که در شکل (۳) مشاهده می‌شود، در فضای مقیاس جدید هر چه به سمت جلوتر حرکت شود و حتی در اکتاوهای بعدی، وضوح و کیفیت تصویر بیشتر می‌شود، ولی از یک نقطه به بعد کم‌کم وضوح تصاویر کاهش می‌یابد به‌نحوی‌که پس از این‌که تعداد اکتاوها از حدی بیشتر شود تنها تصاویر سیاه باقی می‌ماند به همین دلیل است که تعداد اکتاوهای زیاد فایده چندانی ندارد [۱۰].

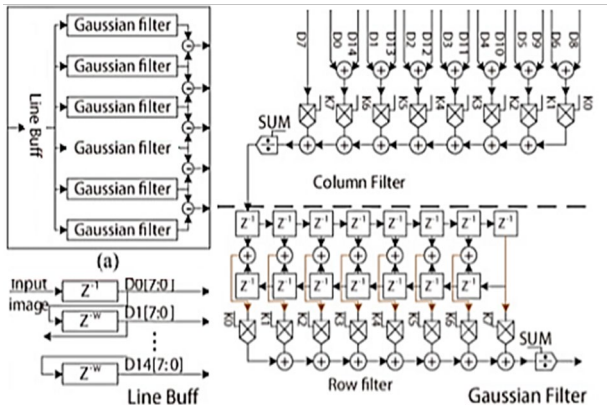


شکل (۳). تصاویر گاوسین و تفاضلات گاوسین این تصاویر

۳- بررسی و مقایسه معماری‌های ارائه‌شده

با توجه به این موضوع که فیلتر گاوسین دو بعدی جدایی-پذیر و متقارن است [۱۱]، Zhong و همکارانش یک روش کارآمد را در [۱۲] پیشنهاد نموده‌اند. در این روش، کانولوشن دوبعدی ابتدا توسط کانوالو با یک فیلتر گاوسین یک بعدی در

جهت افقی و سپس کانوالو با فیلتر گاوسین یک‌بعدی در جهت عمودی صورت می‌گیرد. این روش در شکل (۴) از [۱۲] نشان داده شده است. این روش به لحاظ محاسباتی بسیار کارآمدتر از روش کانولوشن دوبعدی سنتی است زیرا روش کانولوشن دوبعدی سنتی شامل ۲۲۵ ضرب کننده و ۲۲۴ جمع‌کننده برای هر فیلتر گاوسین ۱۵*۱۵ است. علاوه بر این، در روش موردنظر به‌منظور کاهش تعداد ضرب کننده، عملیات ضرب برای مجموع پیکسل-های دارای ضریب مشابه باهم انجام می‌شود. در روش پیشنهادی آن‌ها ۱۶ ضرب کننده و ۲۸ جمع‌کننده و ۱۴ بافر برای هر فیلتر گاوسین ۱۵*۱۵ در نظر گرفته می‌شود. در این معماری تعداد منابع منطقی کاهش داده‌شده است بااین‌حال منابع حافظه در معماری آن‌ها همچنان زیاد و پرهزینه است، چون این معماری ابتدا یک فیلترینگ افقی و سپس یک فیلترینگ عمودی را اجرا می‌نماید [۱۳]. در مقابل یک معماری جدید در [۱۴] به‌صورت شکل (۴)، پیشنهاد شده است. در این معماری ابتدا یک فیلترینگ عمودی و سپس یک فیلترینگ افقی اجرا شده است. درنتیجه بافرهای هر فیلتر گاوسین دو بعدی در هر اکتاو می-توانند از یک بافر خط [1] مشترک استفاده کنند.



شکل (۴). معماری پیشنهادشده برای عملیات تفاضل گاوسین در مرجع [۱۴]

همان‌طوری که مشاهده می‌شود از بافر خط استفاده شده است؛ زیرا ماژول DoG به طور مستقیم توسط جریان تصویر از رابط دوربین هدایت می‌شود و عملیات فیلتر گاوسین دوبعدی و تفریق تصاویر را انجام می‌دهد. به‌منظور خواندن اطلاعات دوربین، در ورودی نمی‌توان از رجیستر استفاده کرد؛ چون حجم داده بالایی وجود دارد که به مقدار آن‌ها حافظه رجیستر برای ذخیره-سازی موجود نیست و ناچار باید از RAM [2] استفاده شود. به-منظور سرعت‌بخشیدن به پردازش از بافر خط استفاده می‌شود که هم‌زمان با گرفتن و ذخیره‌سازی داده‌های ورودی، پردازش آن‌ها هم شروع می‌شود. به‌عبارت‌دیگر بافر خط ستون‌های تصویر را برای ورودی دادن به فیلتر بافر می‌کند و برای پیاده‌سازی تأخیرها در بافر خط چون طول پنجره و تعداد تأخیرها زیاد است از منابع حافظه در داخل FPGA [3] استفاده می‌شود.

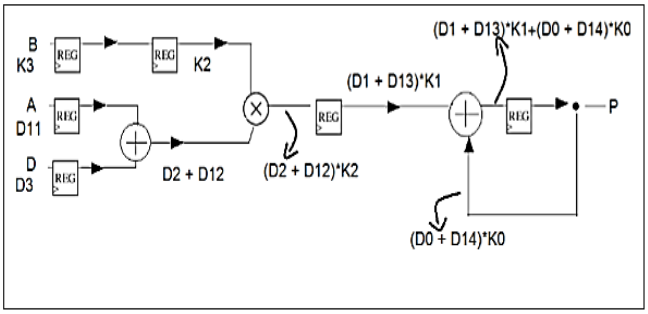
۳-۱- نسبت کلاک مدار به نرخ داده و کاهش منابع استفاده‌شده

نسبت کلاک مدار به نرخ داده تعیین‌کننده میزان کاهش منابع مصرفی FPGA، با رعایت حفظ سرعت مدار است. برای رسیدن به حداکثر سرعت در انجام محاسبات، می‌توان از معماری‌های کاملاً موازی [4] و کاملاً خط لوله‌ای [5] استفاده نمود. در معماری پیشنهادشده در [۱۴]که در شکل (۴) نشان‌داده‌شده است، برای محاسبه فیلتر گاوسین با فیلتر ۱۵*۱۵، به ۱۶ ضرب‌کننده و ۱۴ جمع‌کننده نیاز داریم؛ زیرا به دلیل تقارن در فیلترهای افقی و عمودی برای فیلتر افقی به ۸ ضرب‌کننده و ۷ جمع‌کننده و برای فیلتر عمودی نیز به ۸ ضرب‌کننده و ۷ جمع‌کننده نیاز خواهیم داشت. یعنی برای پیاده‌سازی کل DoG در SIFT که در بخش قبل بررسی گردید به ۱۹۲ ضرب‌کننده و ۱۶۸ جمع‌کننده نیاز داریم. اگر این ضرب‌ها و جمع‌ها بدون استفاده از واحد DSP48 انجام گیرد فرکانس کلاک به‌شدت پایین می‌آید. به طور مثال برای جمع دو مقدار ۴۰ بیتی بدون استفاده از DSP48 به فرکانس 25 MHz خواهیم رسید و بنابراین امکان دستیابی به فرکانس 150 MHz تا 200 MHz وجود ندارد. پس برای رسیدن به حداکثر سرعت پردازش در پیاده‌سازی فیلتر گاوسین ۱۵*۱۵ مجبور به استفاده از ۱۹۲ واحد DSP48 هستیم که این تعداد DSP48 هزینه پیاده‌سازی معماری را به‌شدت بالا می‌برد. اما اگر فرکانس کلاک مدار حداقل ۸ برابر بیشتر از نرخ داده باشد، می‌توان برای پیاده‌سازی هر یک از دو فیلتر عمودی و افقی تنها از یک DSP48 استفاده کرد که با انجام این عملیات تعداد واحدهای DSP48 مصرفی از ۱۹۲ به ۲۴ واحد کاهش می‌یابد. به‌عبارت‌دیگر برای پیاده‌سازی هر فیلتر گاوسین تنها به ۲ واحد DSP48 نیاز داریم و این در حالی است که در [۱۴]به‌جای ۱۹۲ واحد DSP48 از ۵۲ واحد DSP48 استفاده شده است. بنابراین در معماری پیشنهادی نسبت به [۱۴]به تعداد ۲۸ واحد DSP48 کم‌تری استفاده شده است.

اگر تصویر HD [6] با نرخ 60 Fps [7] به ورودی سیستم اعمال شود، درواقع نرخ داده در حدود 12.5 MHz خواهد بود. با در

[1] Line Buffer
[2] Random Access Memory
[3] Field Programmable Gate Array
[4] Fully Parallel
[5] Fully Pipeline
[6] High Definition
[7] Frame per second

نظر گرفتن نرخ پردازشی حدود 150 MHz برای FPGA، می‌توان گفت که در هر ۱۲ کلاک مدار یک داده جدید وارد سیستم می‌شود. بنابراین برای اعمال فیلتر عمودی روی هر یک از داده‌ها به‌اندازه ۱۲ کلاک زمان داریم که با در نظر گرفتن یک کلاک برای محاسبه ضرب و جمع می‌توان به‌جای ۸ واحد DSP48 در این مرحله از یک واحد DSP48 استفاده کرد. همین منطق در مورد فیلتر افقی نیز وجود دارد. نکته‌ای که نیاز به توضیح دارد این است که وقتی به واحد DSP48 ورودی می‌دهیم ممکن است به‌طور مثال ۴ کلاک طول بکشد تا خروجی را حساب کند اما اگر ساختار MAC[1] را به‌صورت خط لوله‌ای قرار دهیم، پس از آماده شدن اولین خروجی، در هر کلاک خروجی خواهیم داشت. می‌توان نشان داد که روند محاسبه هر فیلتر ۱۵*۱۵ قبل از آمدن نمونه بعدی و تغییر در ورودی به‌صورت شکل (۵) است.



شکل (۵). روند محاسبه هر فیلتر ۱۵*۱۵ قبل از آمدن نمونه بعدی

۳-۲- خطای سیستم

در مسیر پردازش داده، دو عامل باعث ایجاد خطا در سیستم می‌شود. یکی از این عوامل، خطای کوانتیزاسیون ناشی از کوانتیزه‌کردن ضرایب در ۱۸ بیت و عامل دیگر خطای تقریب خروجی‌های هر مرحله و کاهش تعداد بیت آن‌ها به ۱۸ بیت است.

اولین خطایی که در محاسبات فیلتر وارد می‌شود ناشی از کوانتیزاسیون ضرایب فیلتر است، زیرا ضرایب فیلتر که به فرمت ممیز شناور[2] و دارای قدرت تفکیک[3] ۲۳ بیت هستند را به‌صورت ۱۸ بیت ممیز ثابت[4] تعریف می‌کنیم که این کاهش دقت، باعث ایجاد خطا در سیستم می‌شود. عامل دوم در عمل خطای زیادی وارد سیستم نمی‌کند. اما در بررسی عامل اول شاید استفاده از ۲۵ بیت برای کوانتیزه‌کردن ضرایب تأثیر مثبت زیادی در کم‌شدن خطا داشته باشد که در عمل بایستی مقایسه‌ای بین این دو عامل صورت گیرد.

در ورودی فیلتر عمودی، حاصل‌جمع مقادیر دو پیکسل ۸ بیتی، یک مقدار ۹ بیتی خواهد داشت و پس از ضرب در ضرایب ۱۸ بیتی متناظر، خروجی ۲۷ بیت خواهد بود. این اعداد ۲۷ بیتی باهم جمع می‌شوند و در هر مرحله جمع، یک بیت به حاصل‌جمع آن‌ها اضافه می‌شود. بنابراین در خروجی فیلتر عمودی یک عدد ۳۴ بیتی خواهیم داشت که برای واردکردن آن به فیلتر افقی، پس از تقریب به ۱۸ بیت تبدیل می‌شود که تمامی این عملیات با فرض استفاده از واحد DSP48 است. در این شرایط امکان سرریز[5] مقادیر میانی حذف شده و خطای اضافی وارد سیستم نمی‌شود، اما خروجی نهایی به‌صورت تقریب‌زدن است. در فیلتر افقی نیز همین روند اتفاق می‌افتد و خروجی نهایی با تقریب، تبدیل به یک عدد ۱۸ بیتی می‌شود.

حال اگر در هرکدام از فیلترهای عمودی و افقی این عملیات گرد کردن را در هر قسمت و برای خروجی هر عملیات انجام دهیم خطای زیادی خواهیم داشت. بنابراین از یک MAC که یک حافظه ۴۸ بیتی است برای ذخیره متغیرهای میانی استفاده می‌شود و با حفظ دقت در طول محاسبات، فقط خروجی نهایی گرد می‌شود. گرد کردن مقدار خروجی فیلتر عمودی و تبدیل آن به ممیز ثابت (۸،۱۰) باعث ایجاد خطا در سیستم می‌شود. درواقع در محاسبات داخلی با درنظرگرفتن تعداد بیت لازم برای متغیرهای میانی دقت حفظ می‌شود؛ اما مقدار خروجی فیلتر عمودی قبل از اعمال به ورودی فیلتر افقی، گرد شده و به ممیز ثابت (۸،۱۰) تبدیل می‌شود. اندازه خروجی فیلتر هرگز بیشتر از ۸ بیت نخواهد شد؛ زیرا همه ضرایب فیلتر دارای مقادیر کوچک‌تر از یک هستند و فقط قسمت اعشار در آن‌ها مقدار دارد؛ بنابراین ضرب اعدادی در حدود ۱۵۶ و ۱۵۷ در ضرایب و سپس مجموع آن‌ها بزرگ‌تر از ۸ بیت نمی‌شود. پس تنها با کم‌کردن تعداد بیت قسمت اعشار در خروجی فیلتر عمودی و همچنین فیلتر افقی خطا وارد سیستم می‌شود.

لازم به ذکر است که مقدار خطایی که در این مرحله وارد سیستم می‌شود از اهمیت برخوردار نمی‌باشد و تشخیص نقاط کلیدی در خروجی الگوریتم به این معنی که چه تعداد از نقاط کلیدی از بین می‌روند و یا اشتباه تشخیص داده می‌شوند اهمیت دارد. به طور مثال ممکن است خروجی الگوریتم ممیز شناور یک پیکسل را نقطه کلیدی تشخیص دهد؛ اما همان پیکسل در خروجی الگوریتم ممیز ثابت به‌عنوان نقطه کلیدی در نظر گرفته نمی‌شود.

[1] Multiply Accumulate
[1] Floating point
[3] Resolution
[4] Fixed point
[5] Overflow

۴- نتایج شبیه‌سازی

ماژول DoG از فراخوانی یک بافر خط و چند بار فراخوانی فیلتر گاوسین تشکیل‌شده است که در انتها، خروجی فیلترهای گاوسین از هم کم می‌شوند و DoG به دست می‌آید.

نکته قابل‌توجه این است که خروجی فیلترهای گاوسین ممیز ثابت (۸،۱۰) و بدون علامت هستند؛ ولی خروجی DoG چون حاصل تفریق است ممیز ثابت (۸،۱۰) و با علامت است.

برای پیاده‌سازی DoG در HLS 2018.3 از زبان برنامه-نویسی C++ استفاده می‌شود که صرفاً بخش مربوط به اکتاو اول پیاده‌سازی شده است. البته برای اکتاو دوم نیز روند به همین صورت است. همچنین کلاک ماژول موردنظر در فرکانس 200 MHz تنظیم می‌شود و با این عمل از یک واحد DSP48 برای هرکدام از فیلترهای عمودی و افقی استفاده می‌شود.

در این روش پیاده‌سازی همان‌گونه که گفته شد، ابتدا یک بافر خط پیاده‌سازی می‌شود و پس از اطمینان از صحت خروجی‌های بافر، این خروجی‌ها از فیلترها عبور می‌کنند و سپس خروجی آن‌ها از هم کم می‌شوند. منابع استفاده‌شده در این روش پیاده‌سازی و معماری [۱۴]در جدول (۱) نشان‌داده‌شده است. باتوجه‌به برآورد انجام‌شده در معماری [۱۴]انتظار داشتیم که تعداد ۱۹۲ واحد DSP48 استفاده شود؛ زیرا معماری [۱۴]شامل ۲ اکتاو و هر اکتاو شامل ۶ فیلتر و هر فیلتر دارای ۸ ضرب‌کننده و فیلترهای افقی و عمودی متقارن است (۲*۸*۶*۲) و همچنین تعداد بسیار کمتری LUT استفاده شود؛ اما به دلیل این‌که پیاده‌سازی به‌صورت بهینه انجام نشده است تعداد واحدهای DSP48 به ۵۲ عدد کاهش‌یافته ولی در مقابل تعداد واحدهای LUT خیلی بیشتر از حد انتظار است. در پیاده‌سازی روش پیشنهادی تعداد واحدهای DSP48 به ۲۴ واحد رسیده است؛ زیرا معماری پیشنهادی شامل ۲ اکتاو و هر اکتاو شامل ۶ فیلتر و هر فیلتر دارای یک ضرب‌کننده و فیلترهای افقی و عمودی متقارن است (۲*۱*۶*۲) و همچنین تعداد واحدهای LUT نیز خیلی کمتر از معماری [۱۴]است. در جدول (۲) منابع استفاده‌شده برای هر یک از فیلترهای گاوسین در روش پیشنهادی ارائه شده است. در این روش، یک واحد DSP48 برای فیلتر عمودی و یک واحد DSP48 برای فیلتر افقی (در مجموع دو واحد) استفاده شده است. نوآوری روش پیشنهادی، کاهش منابع مصرفی FPGA است و کاهش منابع مصرفی به این دلیل است که فرکانس کلاک مدار حداقل ۸ برابر بیشتر از نرخ داده ورودی است. بنابراین می‌توان برای پیاده‌سازی هر یک از دو فیلتر عمودی و افقی تنها از یک واحد DSP48 استفاده کرد. همچنین تعداد واحدهای LUT مورد نیاز نیز خیلی کمتر از معماری [۱۴]است.

جدول (۱). مقایسه منابع استفاده‌شده در [۱۴] و معماری پیشنهادی

	LUT	FF	DSP48	RAM(kbit)
معماری [۱۴]	13982	9836	52	1332
معماری پیشنهادی	4186	4255	24	306

جدول (۲). منابع موردنیاز برای پیاده‌سازی فیلتر گاوسین در معماری پیشنهادی

LUT	FF	DSP48	RAM(kbit)
1460	2100	2	0

برای پیاده‌سازی MAC می‌توان به دو روش عملکرد. در روش اول ورودی واحد DSP48 به‌صورت ممیز ثابت است و گرد کردن مقادیر بر عهده نرم‌افزار HLS قرار داده می‌شود. در روش دوم ورودی واحد DSP48 به‌صورت عدد صحیح است و در خروجی واحد DSP48 مقدار ۴۸ بیت گرد می‌شود. در جدول (۳) تفاوت منابع استفاده‌شده در روش اول و دوم پیاده‌سازی MAC بیان شده است.

جدول (۳). منابع استفاده‌شده در دو روش پیاده‌سازی MAC

	LUT	FF	DSP48	RAM(kbit)
روش اول	825	507	1	0
روش دوم	671	569	1	0

۵- نتیجه‌گیری

همان‌طوری که گفته شد هر چه نسبت کلاک مدار به نرخ داده ورودی بیشتر باشد، تعداد کلاک بیشتری طول می‌کشد تا داده جدیدی وارد سیستم شود. در این شرایط با حفظ سرعت کاری مدار، میزان منابع مصرفی در FPGA کاهش می‌یابد. در معماری پیشنهادی پیاده‌سازی شده باتوجه‌به اینکه کلاک مدار خیلی بیشتر از نرخ داده ورودی است و با استفاده از نوآوری الگوریتم، تعداد واحدهای DSP48 استفاده‌شده نسبت به معماری پیاده‌سازی شده در [۱۴]به میزان یک‌هشتم کاهش‌یافته است. این کاهش منابع مصرفی منجر به کاهش هزینه FPGA مورد نیاز و صرفه اقتصادی خواهد شد. البته معماری به‌کاررفته در [۱۴]به‌صورت کاملاً موازی و کاملاً خط لوله‌ای بوده و قابلیت پردازش ویدئوهای باقدرت تفکیک Full HD@60FPS را دارد که یک مصالحه بین منابع و بیشینه قدرت تفکیک ویدئو وجود دارد.

۶- مراجع

[1]Gholami and M. Okhovat, “Designing Radar and IR Sensors Data Fusion System for Target Tracking in Noise Jamming Conditions,” Electronic and Cyber Defense, vol. 5, no. 3, Serial no. 19, 2017. (In Persian). https://dor.isc.ac/dor/20.1001.1.23224347.1396.5.3.1.4

[2] C. Alabaster, “Pulse Doppler Radar,” Univ. of Cranfield, 2012.

[3] M. Skolnik, “Introduction to Radar System,” New York, McGraw-Hill, 3rd Edition, 2001.

[4] W. R. Dyer “Modern Missile Analysis,” Beltsville, Maryland, 2009.

[5] J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in Proc. 9th IEEE Int. Conf. Comput. Vision, vols. 1–2, pp. 1470–1477, Oct. 2003.

[6] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” IEEE Trans. Pattern Anal Mach Intell, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[7] M. Calonder, V. Lepetit, P. Fua, K. Konolige, J. Bowman, and P. Mihelich, “Compact signatures for high-speed interest point description and matching,” in Proc. 12th IEEE Int Conf Comput Vision, pp. 357–364, Sep.– Oct. 2009.

[8] M. Calonder, V. Lepetit, M. Oezuysal, T. Trzcinski, C. Strecha, and P. Fua, “BRIEF: Computing a local binary descriptor very fast,” IEEE Trans. Pattern Anal Mach Intell, vol. 34, no. 7, pp. 1281–1298, Jul. 2012.

[9] M. I. McCartney, S. Zein-Sabatto, and M. Malkani, “Image registration for sequence of visual images captured by UAV,” in Proc. IEEE Symp. Comput. Intell. Multimedia Signal Vision Process, pp. 91–97, Mar.–Apr. 2009.

[10] C. Harris, “A combined corner and edge detector,” in Proc. 4th Alvey Vision Conf, pp. 147–152, 1988.

[11] S. Mitra, “Digital Signal Processing: A Computer Based Approach,” vol. 1, 3rd ed. New York, NY, USA: McGraw-Hill, 2004.

[12] S. Zhong, J. Wang, L. Yan, L. Kang, and Z. Cao, “A real-time embedded architecture for SIFT,” J Syst Arch., vol. 59, no. 1, pp. 16–29, Jan. 2013.

[13] S. Winder, G. Hua, and M. Brown, “Picking the best DAISY,” in Proc. IEEE Conf Comput Comput. Vis Pattern Recognit., vol. 1–4, pp. 178–185, Jun. 2009.

[14] J. Wang, S. Zhong, L. Yan, and Z. Coa, “An Embedded System-on-Chip Architecture for Real-time Visual Detection and Matching,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 24, no. 3, pp. 525–538, Mar. 2014.