اولین کنفرانس مهندسی فناوری اطلاعات مکانی
دانشگاه صنعتی خواجه نصیرالدین طوسی
دانشکده مهندسی نقشه‌برداری
۲۹ و ۳۰ دی ماه ۱۳۹۴

The 1st National Conference on Geospatial Information Technology
K.N.Toosi University of Technology
Faculty of Geomatics Engineering
19 - 20 January 2016

موازی سازی سخت‌افزاری الگوریتم ISODATA

جواد شعبانی [۱]، رضا شایان [۲]، احسان رحیمی [۳*]

۱- دانشجوی دانشکده مهندسی برق دانشگاه شاهرود
۲- دانشجوی دانشکده مهندسی برق دانشگاه شاهرود
۳- عضو هیات علمی دانشکده مهندسی برق دانشگاه شاهرود

چکیده :

الگوریتم‌های خوشه‌بندی بدون ناظر، تعداد دسته‌ها را به‌صورت خودکار تعیین می‌کنند. این الگوریتم‌ها عموماً دارای بخش‌های تکرار شونده بوده و در تفکیک یا دسته‌بندی و تحلیل تصاویر ماهواره‌ای سنجش از دور یا تصاویر چندطیفی کاربرد زیادی دارند، اما زمان زیادی نیز صرف می‌کنند. لذا پیاده‌سازی سخت‌افزاری آن‌ها با رویکرد موازی‌سازی می‌تواند در بهبود کارایی و سرعت این الگوریتم‌ها بسیار مؤثر باشد. در این مقاله، به بررسی بدنه الگوریتم ISODATA که یکی از الگوریتم‌های خوشه‌بندی بدون ناظر است می‌پردازیم و پس از مشخص کردن نقاط زمان‌بر الگوریتم، آن‌ها را بر روی سخت‌افزار با قابلیت بازپیکربندی، به‌صورت موازی پیاده‌سازی می‌کنیم. نتایج شبیه‌سازی نشان می‌دهد که زمان اجرای این الگوریتم به‌صورت سخت‌افزاری، نسبت به اجرای نرم‌افزاری آن در حدود ۹۰ درصد بهبود می‌یابد. طراحی در پیاده‌سازی سخت‌افزاری، با استفاده از مدارهای ترکیبی و بهینه‌سازی منابع سخت‌افزاری انجام‌شده است

واژه‌های کلیدی : الگوریتم ISODATA، پیاده‌سازی سخت‌افزاری، تصاویر ماهواره‌ای سنجش از دور

نویسنده مکاتبه کننده : احسان رحیمی
آدرس پستی : دانشکده مهندسی برق و رباتیک، شاهرود، سمنان
تلفن :
آدرس پست الکترونیک : erahimi@shahroodut.ac.ir

۱- مقدمه

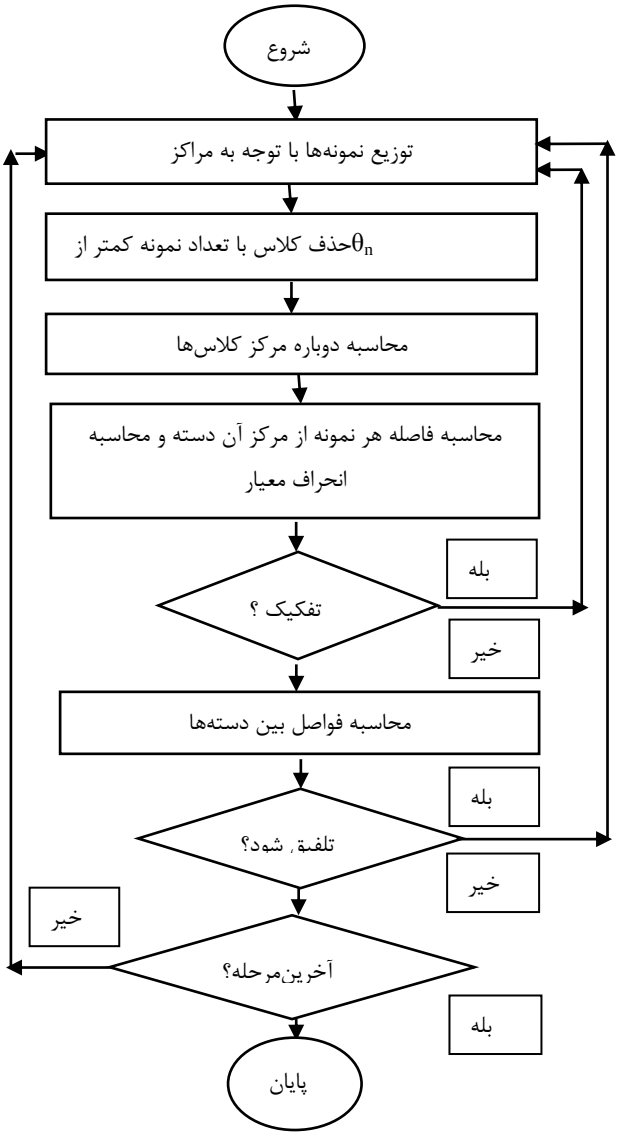
خوشه‌بندی را می‌توان به‌عنوان مهم‌ترین مسئله در یادگیری بدون نظارت در نظر گرفت. در یادگیری بدون نظارت هیچ اطلاعاتی به‌جز داده‌های آموزشی در اختیار یادگیرنده قرار نمی‌گیرد و این یادگیرنده است که باید در درون داده‌ها، ساختاری خاص را جستجو کند [۱]. خوشه‌بندی با یافتن این ساختار، درون مجموعه‌ای از داده‌ها که به گروه خاصی تعلق ندارند سروکار دارد. خوشه به مجموعه‌ای از داده‌ها گفته می‌شود که ازنظر ویژگی‌هایی خاص به یکدیگر شباهت داشته باشند. در خوشه‌بندی سعی می‌شود تا داده‌ها به خوشه‌هایی تقسیم شوند که شباهت بین داده‌های درون هر خوشه حداکثر و شباهت بین داده‌های درون خوشه‌های متفاوت حداقل شود. یکی از پرکاربردترین الگوریتم های خوشه‌بندی بدون نظارت الگوریتم ISODATA است که کاربردهای بسیاری در سامانه‌های نقشه‌برداری و تجزیه و تحلیل نقشه‌های چند طیفی و دسته‌بندی تصاویر سنجیده شده از دور دارد.[۲ و۳]. اما مشکل اینجاست که اجرای این نوع خوشه‌بندی، به علت اینکه الگوریتمی تکرار شونده دارد، در کاربردهای زمان حقیقی که نمونه‌ها، تعداد خوشه‌ها، تعداد دسته‌ها و مرکز دسته‌ها به‌سرعت تغییر می‌کنند، همچنین تصاویر ماهواره‌ای که چندکاناله هستند، تصاویر متحرک، تصاویری که تغییرات زیادی دارند، تصاویر دوربین ها و تصاویر حسگرها، فرایند زمان‌بری است. در حالی که در مسئله خوشه‌بندی، همواره کوتاه بودن زمان اجرای الگوریتم بسیار اهمیت دارد. برای مقابله با این چالش، می‌توان پیاده‌سازی سخت‌افزاری و موازی‌سازی الگوریتم را به کار گرفت. در این پژوهش برای به‌روزرسانی مرکز خوشه‌ها که یکی از مراحل مسیرهای داده است، به تعداد خوشه‌ها واحدهای موازی در نظر گرفته‌شده است. به‌طوری که محاسبه فاصله مرکز دسته‌ها تا پیکسل‌ها به‌صورت همزمان انجام می‌شود.

۲- الگوریتم ISODATA

این الگوریتم با استفاده از فن جداسازی و ادغام در تکرارهای متوالی، تعداد خوشه‌ها را مشخص می‌کند. درواقع این الگوریتم نسخه بهبودیافته الگوریتم K-means است؛ اما قوانینی به آن افزوده شده که برخی چالش‌های K-means را برطرف می‌کند.

۲-۱ - مراحل الگوریتم ISODATA

در ابتدا، کاربر مقادیر آستانه برای پارامترهای الگوریتم تعریف می‌کند. برخی از این پارامترها عبارت اند از: ۱- تعداد خوشه‌های اولیه، ۲- حداکثر تعداد تکرارهای مجاز، ۳- حداکثر تعداد جفت خوشه‌هایی که اجازه ادغام شدن دارند، ۴- حداقل تعداد نمونه های هر خوشه (θ_n) که برای حذف برخی از خوشه‌ها استفاده می شود، ۵- یک مقدار آستانه برای انحراف معیار که برای جداسازی استفاده می‌شود و ۶- فاصله معقول بین یک جفت خوشه که برای عمل ادغام استفاده می‌شود، [۴ و ۵]. شکل (۱) این مراحل را نشان می‌دهد.



شکل(۱) مراحل الگوریتم ISODATA

در آغاز اجرای الگوریتم، از بین کل نمونه‌ها، تعداد K نمونه به‌صورت تصادفی به‌عنوان مراکز اولیه خوشه‌ها انتخاب می‌شود. سپس در مرحله اول هرکدام از نمونه‌ها به نزدیک‌ترین مرکز نسبت داده می‌شود. در مرحله دوم خوشه‌هایی که N_θ را ارضا نمی‌کنند حذف می‌شوند؛ و به تعداد خوشه‌های حذف‌شده، از مقدار K کم می‌شود. در مرحله سوم، مرکز هر خوشه به‌روز می‌شود. سپس، در مرحله چهارم، فاصله میانگین هر نمونه از مرکز خوشه‌اش محاسبه می‌شود. در مرحله پنجم، شرایط برای ادغام و جداسازی بررسی می‌شود و در صورت نیاز این اعمال اتفاق می‌افتد و درنهایت، الگوریتم به مرحله اول بازمی‌گردد و این روال ادامه پیدا می‌کند تا مقادیر آستانه‌ای که در ابتدا تعریف‌شده است، ارضا شوند، [۵ و ۶].

۲-۲- بخش‌های زمان‌بر الگوریتم

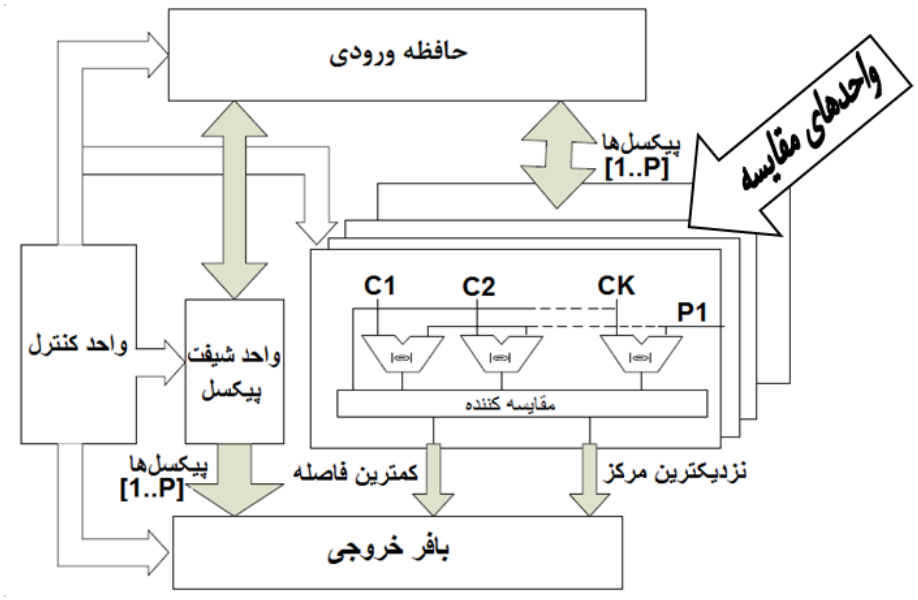
محاسبه فاصله دوبه‌دو، محاسبه انحراف استاندارد و محاسبه فاصله میانگین، بخش‌های زمان‌بر الگوریتم در پیاده‌سازی نرم‌افزاری می‌باشند که در این میان طبق آنچه در مرجع [۵] بررسی شده، محاسبه فاصله دوبه‌دو با صرف حدود ۶۰٪ از کل زمان محاسبات، رتبه اول را به خود اختصاص می‌دهد، (جدول۱).

جدول ۱- متوسط زمان مصرف‌شده برای بخش‌های مختلف الگوریتم که به‌صورت نرم‌افزاری پیاده‌سازی شده است

عملیات	میانگین درصد
محاسبه فاصله دوبه‌دو	٪۲۷/۶۱
محاسبه انحراف استاندارد	٪۲۴/۱۶
محاسبه فاصله میانگین	٪۱۰/۳۱
تقسیم خوشه	٪۳/۳
ادغام خوشه	٪۴۲/۴
دیگر عملیات	٪۴/۴۶

۴- پیاده‌سازی سخت‌افزاری

با توجه به فرایند لازم جهت احراز شرایط برای ادغام و تلفیق دسته‌ها، اجرای نرم‌افزاری آن بهینه‌تر از پیاده‌سازی سخت‌افزاری خواهد بود. برای طراحی قسمت‌های قابل پیاده‌سازی ISODATA به زبان سخت‌افزار، از مسیر داده پیشنهادی در مرجع [۵] که در شکل(۲) نشان داده شده، استفاده شده است.



شکل(۲) مسیر داده استفاده شده برای پیاده‌سازی سخت‌افزاری الگوریتم

مطابق این شکل، پیکسل‌هایی که قرار است محاسبات فاصله روی آن‌ها انجام شود در حافظه ورودی قرار می‌گیرند و با فرمان‌های صادر شده توسط واحدهای شیفت و کنترل پیکسل، به صورت بسته‌های چند پیکسلی به واحدهای مقایسه تحویل داده می‌شوند. سپس مقادیر خروجی این واحدها در اختیار بافر خروجی قرار می‌گیرد. در این پژوهش از سه واحد برای پیاده‌سازی شکل(۲) استفاده شده است: ۱- واحد حافظه، ۲- واحد مقایسه و ۳- واحد بافر خروجی.

۴-۱- واحد حافظه

برای این واحد دو حافظه جداگانه در نظر گرفته‌شده است که یکی از آن‌ها پیکسل‌ها، و دیگری مراکز خوشه‌ها را در خود ذخیره می‌کند. ورودی‌های این واحد شامل یک خط داده n بیتی برای وارد شدن پیکسل‌ها و مرکزها (data_in)، فعال‌ساز خواندن و نوشتن در حافظه‌ی مخصوص پیکسل‌ها (p_rd_en,p_wr_en)، فعال‌ساز خواندن و نوشتن در حافظه‌ی مختص مراکز (c_rd_en,c_wr_en)، بازنشانی (rst) و منبع پالس ساعت(clk) هست. همچنین خروجی‌ها شامل z خط داده n بیتی برای پیکسل‌ها (p_z)، k خط داده n بیتی برای مراکز خوشه (c_k) و سیگنال‌های کنترلی برای تشخیص پر یا خالی بودن حافظه پیکسل (p_full,p_empty) است.

با فعال شدن حافظه مراکز، مقادیر مراکز خوشه‌ها در هر لبه بالارونده پالس ساعت به‌نوبت وارد حافظه می‌شوند و با پر شدن حافظه، بعد از یک لبه بالارونده پالس ساعت خواندن، در k خط داده n بیتی روی خروجی قرار می گیرند. به همین صورت مقادیر مربوط به پیکسل‌ها با فعال شدن حافظه پیکسل، دریافت می‌شوند و در خروجی قرار می‌گیرند. بازنشانی نیز در آغاز کار برای صفر کردن شمارنده‌های خط آدرس حافظه به کار می‌رود. تعداد خانه‌های حافظه با تعداد پیکسل‌هایی که قرار است پردازش موازی شوند برابر است.

۴-۲- واحد مقایسه

ورودی‌های این واحد شامل یک خط داده n بیتی برای پیکسل(p) و k خط داده n بیتی برای مراکز خوشه‌ها (c)، و خروجی‌ها شامل خط داده n بیتی برای نزدیک ترین مرکز (cc) و کمترین فاصله (cd) است. برای به دست آوردن فاصله پیکسل‌ها از یکدیگر از عمل تفریق استفاده شده است که پیاده‌سازی آن به زبان سخت‌افزار نسبت به پیاده‌سازی سخت‌افزاری فرایند محاسبه فاصله اقلیدسی ساده‌تر است. به همین علت واحد مقایسه به تعداد مراکز خوشه‌ها از تفریق کننده برخوردار است. همچنین این واحد کاملاً با استفاده از مدارهای ترکیبی طراحی شده است تا تغییرات ورودی‌ها، با سرعتی مناسب در خروجی لحاظ شود و زمان پردازش حقیقی کاهش یابد. در این واحد، فاصله هر پیکسل از همه مرکزها محاسبه می‌شود و مقدار کمترین فاصله و نزدیک‌ترین مرکز، در خروجی آن قرار می‌گیرد. به عبارتی این واحد مشخص می کند که هر پیکسل به کدام مرکز نزدیک‌تر است و بدین ترتیب، خوشه‌ها تعیین می‌شوند. برای تحقق پردازش موازی، به ازای هر پیکسل به یک واحد مقایسه احتیاج است. هر چه تعداد این واحدها بیشتر باشد سرعت اجرای الگوریتم نیز بیشتر خواهد شد.

۴-۳- واحد بافر خروجی

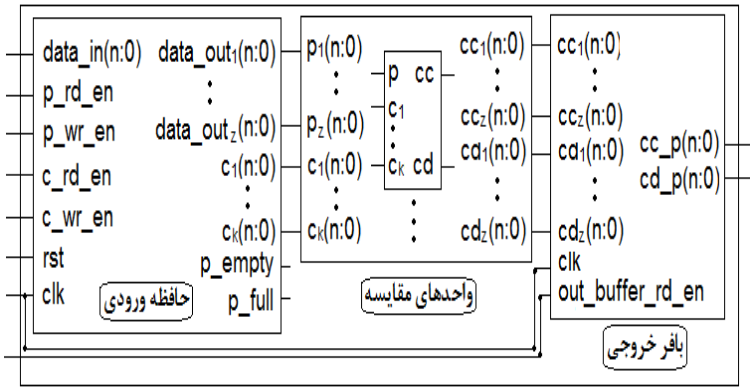
ورودی‌های این واحد شامل فعال‌ساز خواندن بافر (out_put_rd_en)، منبع پالس ساعت(clk) و 2^z خط داده n بیتی، برای مقادیری که از پردازش موازی پیکسل‌ها حاصل شده، و خروجی شامل دو خط داده n بیتی برای قرارگیری ترتیبی نزدیک‌ترین مرکز و کمترین فاصله است. در صورت فعال بودن بافر، با هر لبه بالارونده پالس ساعت به ترتیب مقادیر نزدیک‌ترین مرکز و کمترین فاصله، در خروجی قرار می‌گیرد.

۴-۴- ساخت واحد نهایی

پس از اتصال واحدهای مطرح شده به یکدیگر، پیاده‌سازی الگوریتم ISODATA به زبان سخت‌افزار مطابق با مسیر داده پیشنهاد شده در شکل(۲)، به پایان می‌رسد. طراحی به گونه‌ای صورت گرفته که ورودی‌های فعال‌ساز واحدها، نقش واحد کنترل را ایفا می‌کنند و عمل شیفت در هر بار پر و خالی شدن بافر پیکسل انجام می‌شود. درنتیجه احتیاجی به طراحی واحدهای کنترل و شیفت به‌صورت مجزا، وجود ندارد. برای گرفتن پاسخ سریع‌تر، می‌توان عمل خواندن از بافر خروجی و نوشتن در حافظه ورودی را به‌طور همزمان انجام داد.

۵-شبیه‌سازی سخت‌افزار طراحی شده

سخت‌افزار مشخص شده در شکل(۳) در نرم‌افزار modelsim شبیه‌سازی شده است. این شبیه‌سازی به‌عنوان نمونه برای ده پیکسل و چهار مرکز خوشه ثابت، به عرض داده هشت بیت انجام شده که نتیجه آن در شکل(۴) قابل ملاحظه است.



شکل(۳) بلوک سخت‌افزار واحد نهایی

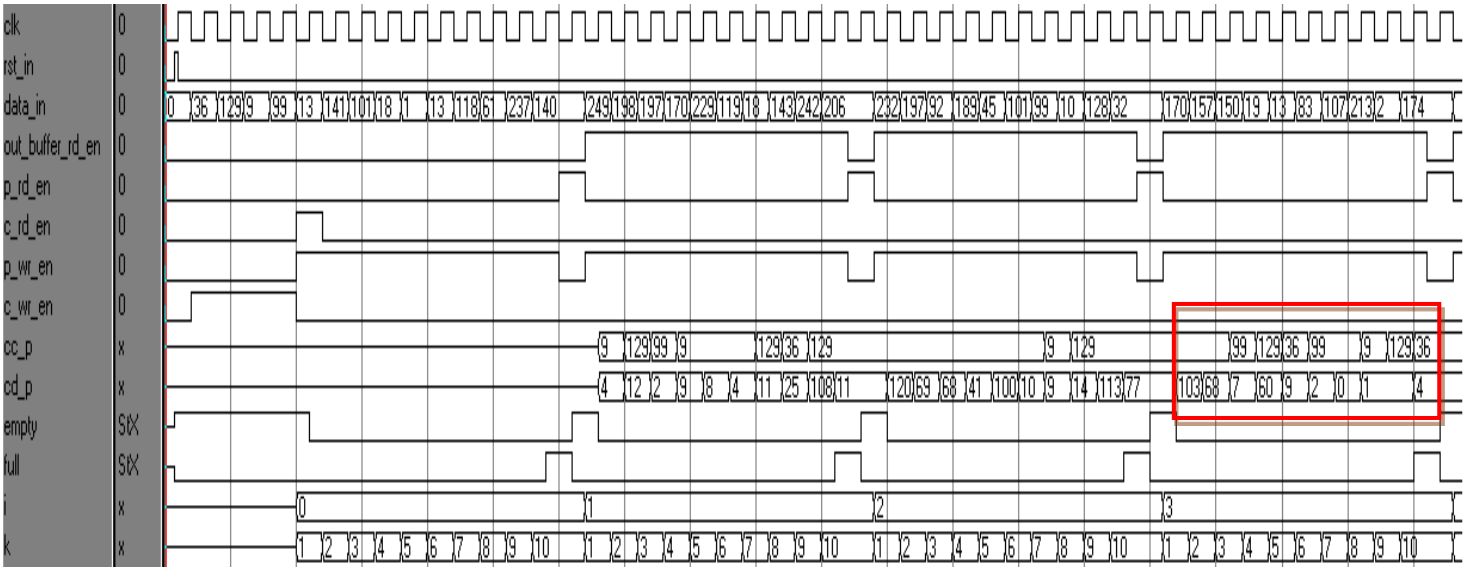
در این شکل در ردیف مربوط به مقادیر data_in اعداد ۳۶، ۱۲۹، ۹ و ۹۹ به ترتیب بیانگر مراکز ثابت C_1، C_2، C_3 و C_4 و سایر اعداد، گروه‌های ده‌تایی پیکسل‌های وارد شده به حافظه را نشان می‌دهند. در این شبیه‌سازی بعد از هر یازده لبه بالارونده ساعت، نتیجه پردازش موازی روی پیکسل‌ها قابل استخراج است. برای آزمودن نتیجه شبیه‌سازی، فاصله سومین گروه ده‌تایی از پیکسل‌های وارد شده به واحد حافظه را از مراکز خوشه‌ها محاسبه کردیم. نتایج در جدول(۲) نمایش داده شده است. از طرفی حاصل شبیه‌سازی این گروه در شکل (۴) توسط کادر قرمزرنگ مشخص‌شده است. مقادیر هاشور خورده در جدول(۲) کمترین فاصله‌ها را نشان می‌دهند. با مقایسه این اعداد با نتیجه شبیه‌سازی، ملاحظه می‌شود که سخت‌افزار طراحی شده به‌درستی کار می‌کند. لازم به ذکر است که مقادیر تمام داده‌ها به‌صورت تصادفی انتخاب‌شده است. توصیف این سخت‌افزار در نرم‌افزار Xilinx ISE بر روی تراشه vlx254 از خانواده virtex4 انجام‌شده است. با توجه به جدول(۳) میزان استفاده شده از سخت‌افزار در دسترس کمتر از ۱۰ درصد است. همچنین تعداد پایه‌های استفاده شده در این شبیه‌سازی، ۳۳ عدد است. در حالت کلی، تعداد پایه‌های مورد نیاز برای ایجاد سخت‌افزار با داده‌هایی به عرض n بیت برابر $(3n+9)$ پایه است. ماکزیمم فرکانس کاری در این شبیه‌سازی ۱۱۰ مگاهرتز به دست آمده است که در صورت بهینه‌سازی سخت‌افزار می‌توان به فرکانس‌های کاری بیشتر رسید.

جدول۲- مقادیر داده ورودی و نتیجه محاسبه فاصله هر پیکسل از همه مراکز

	مرکز خوشه			
پیکسل	c_1	c_2	c_3	c_4
$p_1 = ۲۳۲$	۱۳۳	۲۲۳	۱۰۳	۱۹۶
$p_2 = ۱۹۷$	۹۸	۱۸۸	۶۸	۱۶۱
$p_3 = ۹۲$	۷	۸۳	۳۷	۵۶
$p_4 = ۱۸۹$	۹۰	۱۸۰	۶۰	۱۵۳
$p_5 = ۴۵$	۵۴	۳۶	۸۴	۹
$p_6 = ۱۰۱$	۲	۹۲	۲۸	۶۵
$p_7 = ۹۹$	۰	۹۰	۳۰	۶۳
$p_8 = ۱۰$	۸۹	۱	۱۱۹	۲۶
$p_9 = ۱۲۸$	۲۹	۱۱۹	۱	۹۲
$p_{10} = ۳۲$	۶۷	۲۳	۹۷	۴

جدول۳- میزان استفاده شده از سخت‌افزار

Logic Utilization	Used	Available	Utilization
Number of Slices:	1121	10752	10%
Number of Slice Flip Flops:	374	21504	1%
Number of 4 input LUTs:	2141	21504	9%
Number of bonded IOBs:	33	242	13%
Number of GCLKs:	1	32	3%



شکل۴- نتیجه شبیه‌سازی برای ده پیکسل و چهار مرکز خوشه ثابت

۶-سرعت پردازش سخت‌افزار نسبت به نرم‌افزار

فاصله هر پیکسل باید از مراکز خوشه محاسبه شود که در این صورت به $p \times k$ عمل تفریق نیاز است. برای تشخیص کمترین فاصله و نزدیک ترین مرکز نیز حداقل به $p \times k$ عمل نیاز است. بدیهی است که در پیاده سازی نرم‌افزاری، هر یک از اعمال فوق بیشتر از یک لبه پالس ساعت نیاز دارند و این در حالی است که در پیاده‌سازی سخت‌افزاری، تنها با لبه ساعت تمام اعمال فوق انجام می‌شود. با در نظر گرفتن بدترین حالت، به جای هر $p \times k$ ۲ لبه ساعت که در پیاده سازی نرم‌افزاری مصرف می‌شود به یک لبه ساعت در پیاده‌سازی سخت‌افزاری برای محاسبه فاصله دوبه‌دو نیاز است.

۷-نتیجه‌گیری

در این مقاله قسمت‌های زمان‌بر الگوریتم ISODATA بررسی و واحد محاسبه فاصله دوبه‌دو پیکسل‌ها از مراکز خوشه به‌صورت سخت-افزاری پیاده‌سازی شد. در صورتی که در این سخت‌افزار، z پیکسل و k مرکز خوشه به‌صورت موازی مورد پردازش قرار گیرند با هر z+k+1 لبه بالارونده ساعت حاصل فاصله دوبه‌دو را برای تمام پیکسل‌ها می‌توان به دست آورد. پیاده‌سازی این سخت‌افزار بر روی تراشه 4vlx25 از خانواده Virtex4 انجام‌شده است که در شبیه‌سازی صورت گرفته، حدود ۱۰ درصد از حجم سخت‌افزار موجود مورد استفاده قرار گرفت. در صورت استفاده تمام حجم سخت‌افزار و انجام بهینه‌سازی‌های لازم برای پردازش موازی پیکسل‌های بیشتر، پیش‌بینی می‌شود که زمان صرف شده برای تشخیص فاصله دوبه‌دو در الگوریتم به کمتر از ۵ درصد در مقابل ۶۰ درصد، در پیاده‌سازی نرم‌افزاری کاهش پیدا کند.

مراجع

[1] F. Keller, "Clustering", Computer University Saarlandes, Tutorial Slides.

[2] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Classification And Scene Analysis", John Wiley & Sons, 2000

[3] M. Poostchi, M. Analooee,"Investigation of ISODATA Challenges by Decrease of Time Level of Associated Phases", 7th Conference on Fussy Systems, Mashhad,1386

[4] Memarsadeghi, N., Netanyahu, N.S., LeMoigne, J., 2006 A Fast Implementation of the ISODATA Clustering Algorithm, International Journal of Computational Geometry and Applications.

[5] E. Rahimi, Sh. B.Shokouhi, A. sadr "A Parallelized and Pipelined Datapath to Implement ISODATA Algorithm for Rosette Scan Images on a Reconfigurable Hardware",IEEE International Conference on Granular Computing, USA, 200

[6] Earl Gose,1996. Pattern Recognition and Image analysis. Prentice Hall

[7] H. Sadeghian, Sh. B. Shokouhi, "Hardware Implementation of ISODATA for Rosette IR Seeker", 8th Conference of Aerospace Association, Malek Ashtar Univ. Tehran, 1387.

Hardware Parallelization of the ISODATA Algorithm

J. Shabani[1], R. Shayan[2] and E. Rahimi[3*]

[1,2,3]School of Electrical Engineering, Shahrood University

[*]erahimi@shahroodut.ac.ir

Abstract

Unsupervised classical algorithms have a potential application in classification of multichannel images produced by remote sensing and geographical information systems. These Algorithms are iterative, and the time consumed is considerable for a multichannel map in the case of software platform. In this paper we implement the ISODATA algorithm on a parallel reconfigurable hardware in a combinational logic data-path approach. The simulation results confirm a 90% improvement of the consuming time of the algorithm compared to software implementations.