ISSN: 2821-1553 htpps://www.jasd.khadu.ir

فصلنامه علمی ((دفاع هوافضایی))

دوره ۱، شماره ۳، آذر ۱۴۰۱

مقاله پژوهشی

استفاده از ضرایب کرولیشن جهت ارائه یک الگوریتم بهینه تطبیق الگو، تغییرناپذیر نسبت به مقیاس و زاویه با پیاده‌سازی بر روی FPGA

مجید زارعی[۱] ، علی جاهد سراوانی[۱] ، قادر محمدی آغداش[۲] ، اسماعیل دستجردی[۲]

۱- استادیار، دانشکده مهندسی برق، دانشگاه پدافند هوایی خاتم‌الانبیاء(ص)، تهران، ایران

۲- دانشکده مهندسی برق، دانشگاه پدافند هوایی خاتم‌الانبیاء(ص)، تهران، ایران

اطلاعات مقاله

تاریخ پذیرش: ۱۴۰۱/۰۵ /۳۰

تاریخ دریافت: ۱۴۰۰/۱۱/۲۲

کلمات کلیدی:

تطبیق الگو، تغییرناپذیری نسبت به مقیاس و زاویه، ردیابی، ضرایب کرولیشن.

نویسنده مسئول:

مجید زارعی

ایمیل:

majidzarie@yahoo.com

چکیده

امروزه در سیستم‌های ردیاب مبتنی بر تصویر، جهت شناسایی و تعقیب اهداف نظامی به‌طور گسترده‌ای از تکنیک‌های تطبیق الگو استفاده می‌شود. سیستم‌های فوق به دلیل پسیو بودن، در برابر تکنیک‌های جنگ الکترونیک رایج مقاوم هستند. یکی از مشکلات استفاده از الگوریتم‌های تطبیق الگو، کند بودن آن‌ها بخصوص در مواقع بروز چرخش و تغییر مقیاس در اهداف نسبت به الگوی از قبل تعیین‌شده می‌باشد. در این مقاله الگوریتمی بهینه‌شده جهت انجام تطبیق الگو مبتنی بر ضرایب کرولیشن، تغییرناپذیر به مقیاس و زاویه در تصاویر مقیاس خاکستری بیان شده است. الگوریتم brute force تطبیق الگو بین الگوی (**Q**)و تصویر(**A**) را از طریق ایجاد حالت‌های مختلف الگو در مقیاس و زاویه‌های مشخص انجام می‌دهد. این الگوریتم بیش‌ازحد طولانی و زمان‌بر بوده، جنبه عملی ندارد. الگوریتم بهینه‌شده شامل ۳ فیلتر متوالی مقیاس، زاویه و تطبیق الگو می‌باشد. که در آن‌ها به ترتیب احتمال مقیاس، زاویه چرخیده شده برای هر پیکسل و درنهایت نقطه تطبیق الگو به دست می‌آید. این الگوریتم برخلاف brute force شتاب قابل‌توجهی به جستجو داده و نزدیک به۴۰۰ بار سریع‌تر از آن عمل می‌کند. با پیاده‌سازی سخت‌افزاری بر روی FPGA می‌توان عملیات شناسایی و تعقیب اهداف نظامی را با سرعت ۱۰ فریم در ثانیه انجام داد.

استناد به مقاله: مجید زارعی ، علی جاهد سراوانی ، قادر محمدی آغداش ، اسماعیل دستجردی، استفاده از ضرایب کرولیشن جهت ارائه یک الگوریتم بهینه تطبیق الگو، تغییرناپذیر نسبت به مقیاس و زاویه با پیاده‌سازی برروی FPGA ، مجله علمی پژوهشی دفاع هوافضایی دوره ۱، شماره ۳، آذر ۱۴۰۱.

ISSN:2821-1588 htpps://www.jasd.khadu.ir

Journal of Airspace Defense

Research Paper

Use of Correlation Coefficients to Provide an Optimal Algorithm of Template Matching, Invariant to Scale and Angle with Implementation on FPGA

Majid Zareie[1] , Ali Jahed Saravani[1] , Ghader Mohammadi Aghdash[2] , Esmaeel Dastjerdi[2]

[1] Assistant Professor, Electrical Engineering Department, khatam al-anbia (pbuh) University, Tehran, Iran

[2] Electrical Engineering Department, khatam al-anbia (pbuh) University, Tehran, Iran

Article Information

Accepted: 1401/05/30

Recceived:1400/11/22

Keywords:

Template Matching, Invariant to Scale and Angle, Tracing, Correlation Coefficients..

Corresponding anuthor:
Majid Zareie
Email:
majidzarie@yahoo.com

Abstract

Template matching techniques are widely used today in image-based tracking systems to identify and track military targets. Due to their passivity, these systems are resistant to common electronic warfare techniques. One of the problems with using template matching algorithms is that they are slow, especially when rotation and scaling occur in targets relative to a predetermined pattern. In this paper, we perform an optimized algorithm for grayscale template-matching based on correlation coefficients which is invariant to scale and angle. The 'brute force' algorithm performs template-matching between the image to analyze and the template query shape rotated by specific angle, translated to specific scale. This takes too long and thus is no practical. The optimized algorithm includes three cascaded filters for scale, angle and template matching which results in probability of scaling, rotated angle for each pixel and point of template-matching, respectively. This algorithm accelerates searching and is 400 times faster than 'Brute Force' algorithm. By implementing hardware on the FPGA, it is possible to identify and track military targets at a speed of 10 frames per second.

HOW TO CITE: Majid Zareie , Ali Jahed Saravani , Ghader Mohammadi Aghdash , Esmaeel Dastjerdi, Use of Correlation Coefficients to Provide an Optimal Algorithm of Template Matching, Invariant to Scale and Angle with Implementation on FPGA, Journal of Airspace Defense, Vol. 1, No, 3, 1401.

۱. مقدمه

تطبیق الگو روشی است برای تعیین نمودن محلی از تصویر مورد پردازش، که شباهت قابل قبولی با الگوی موردنظر داشته باشد. امروزه از روش‌های تطبیق الگو جهت ردیابی، شناسایی و تعقیب اهداف نظامی به‌طور گسترده استفاده می‌شود [۱-۴]. به‌علاوه سیستم‌های مبتنی بر تصویر به دلیل پسیو بودن، در برابر تکنیک‌های جنگ الکترونیک رایج مقام هستند. هدف از مقاله حاضر ارائه الگوریتمی بهینه‌شده از تطبیق الگو مبتنی بر ضرایب کرولیشن، تغییرناپذیر به مقیاس و چرخش می‌باشد. بدین معنی که ممکن است الگوی Q در تصویر A دچار چرخش و تغییر مقیاس شده باشد. الگوریتم بهینه‌شده با استفاده مستقیم از تصاویر خاکستری، بدون انجام هیچ‌گونه ساده‌سازی قبلی از قبیل تشخیص لبه‌ها، آشکارسازی نقاط موردعلاقه، قطعه‌بندی و باینری‌کردن تصاویر، عمل تطبیق الگو را با نهایت دقت انجام می‌دهد. چون این ساده‌سازی‌ها ازیک‌طرف ممکن است اطلاعاتی باارزش از تصاویر خاکستری را حذف کند و از طرفی دیگر این قبیل فرآیندها حساس به نویز و مستعد به خطا می‌باشند و باعث می‌شوند تطبیق الگو به‌درستی انجام نشود. در کاربردهای نظامی مربوط به شناسایی، سرعت و دقت در انجام عملیات دو رکن مهم و قابل‌ملاحظه هستند که الگوریتم بهینه‌شده فرآیند تطبیق الگو را با دقت بالا و کمترین زمان انجام می‌دهد [۱،۴].

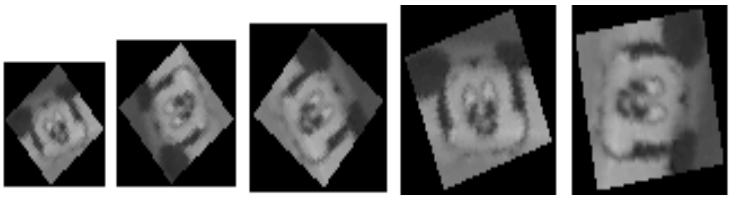
الگوریتم پایهٔ brute force عمل تطبیق بین الگوی Q و تصویر A را به این صورت انجام می‌دهد. ابتدا الگوی Q را تحت مقیاس‌های مشخص تغییر اندازه داده و الگوهای تغییریافته را تحت زاویه-های مشخص می‌چرخاند [۵] سپس عمل تطبیق را بین الگوهای تولیدشده و تصویر مورد آنالیز انجام می‌دهد. ازآنجایی‌که این روش بیش‌ازحد طولانی و زمان‌بر بوده، کمتر جنبه عملی دارد. در تعدادی از الگوریتم‌های تطبیق الگو فقط جنبه تغییرناپذیری به چرخش توجه شده است [۶،۷]. در برخی دیگر از الگوریتم‌ها که تغییرناپذیر به چرخش بوده، با انجام یک سری عملیات جداسازی در هر بخش از تصویر و باینری‌کردن آن، تغییرناپذیری به مقیاس هم لحاظ شده است [۸،۹]. بنا به دلایل گفته‌شده با این قبیل ساده‌سازی‌ها (قطعه‌بندی، باینری‌کردن) تطبیق الگو به‌درستی انجام نمی‌شود.

الگوریتم بهینه‌شده شامل ۳ فیلتر متوالی مقیاس، زاویه و تطبیق الگو می‌باشد که در آن‌ها به ترتیب احتمال مقیاس، زاویه چرخیده شده برای هر پیکسل و درنهایت نقطه تطبیق الگو به دست می‌آید. در هر مرحله پیکسل‌هایی که شانسی در تطبیق الگو ندارند کنار گذاشته شده تا مانع از انجام عملیات مجدد بر روی آن‌ها شوند [۱۰]. در ادامه الگوریتم brute force به‌عنوان الگوریتم پایه موردبررسی قرار گرفته و الگوریتم بهینه‌شده به همراه نتایج شبیه‌سازی، پیاده‌سازی سخت‌افزاری بر روی FPGA و نتیجه‌گیری به ترتیب در بخش‌های بعدی ارائه شده است.

۲. الگوریتم brute force

الگوریتم پایه brute force عمل تطبیق بین الگوی Q و تصویر A را به این صورت انجام می‌دهد. در ابتدا الگوی Q را تحت مقیاس‌های مشخص به طور مثال n=6 (s_1=0.6, s_2=0.7, … s_6=1.1)

تغییر اندازه داده [۱۱]، سپس الگوهای تغییریافته را تحت زاویه‌های مشخص به طور مثال $m=36$ $(\alpha_1=0, \alpha_2=10, \ldots \alpha_{36}=350)$ می‌چرخاند. شکل ۱ تعدادی از ۲۱۶ الگو ایجادشده را نشان می‌دهد.

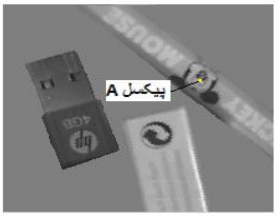


شکل ۱: تعدادی از الگوهای تغییر مقیاس و تغییر زاویه یافته

در ادامه با جابه‌جایی تک‌تک الگوها در سرتاسر تصویر A و محاسبه میزان شباهت آن‌ها به‌وسیله ضرایب کرولیشن از طریق معادله ۱ تطبیق الگو را انجام می‌دهیم [۱۲-۱۴]:

$$corr(x,y)=\frac{\sum_s\sum_t\left(Q(s,t)-\bar{Q}\right)\left(A(x+s,y+t)-\bar{A}\right)}{\sqrt{\sum_s\sum_t\left(Q(s,t)-\bar{Q}\right)^2\sum_s\sum_t\left(A(x+s,y+t)-\bar{A}\right)^2}} \qquad (۱)$$

پارامترهای معادله ۱ به‌صورت زیر تعریف می‌شود. الگو Q ، میانگین پیکسل‌های الگو $\bar{Q}$، قسمت انتخاب‌شده به‌اندازه الگو از تصویر A، میانگین پیکسل‌های قسمت انتخاب‌شده $\bar{A}$، اندازه سطر و ستون الگو Q (s , t)، مختصات پیکسل تصویر A (x , y) و مقدار corr بین [۱, ۱-] می‌باشد. هرچقدر این عدد به یک نزدیک‌تر باشد وابستگی بیشتری بین الگو Q و قسمت انتخاب‌شده از تصویر A را نشان می‌دهد و می‌تواند محل مناسبی برای فرآیند تطبیق الگو باشد. برای تک‌تک پیکسل‌های تصویر A به ازای تمام الگوهای ایجادشده، معادله(۱) را تکرار می‌کنیم. با معین کردن یک مقدار آستانه به طور مثال $t = 0.85$، پیکسل‌هایی از تصویر A که مقدار corr آن‌ها از t بیشتر باشد به‌عنوان پیکسل کاندید ذخیره می‌کنیم. درنهایت مقدار ماکزیمم کرولیشن‌ها، محل تطبیق الگو می‌باشد. در شکل ۲ نمونه‌ای از پیاده‌سازی این الگوریتم که نقطه تطبیق الگو با پیکسل A به رنگ زرد نشان داده شده است.



شکل ۲: خروجی الگوریتم brute force

۳. الگوریتم بهینه‌شده

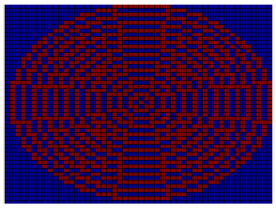
در این بخش الگوریتم بهینه که شامل ۳ فیلتر متوالی مقیاس، زاویه و تطبیق الگو می‌باشد با جزئیات کامل شرح می‌دهیم.

۳-۱ فیلتر مقیاس

در این فیلتر از یک سری طرح‌های دایره‌ای روی الگوی Q و تصویر A استفاده می‌کنیم تا در انتهای این فیلتر از بین پیکسل‌های تصویر A، پیکسل‌های کاندید نوع اول با احتمال مقیاس مربوطه را به دست آوریم. قبل از شروع پیاده‌سازی این فیلتر لازم است نمونه‌برداری دایره‌ای $C_B(x,y,r)$ را با توجه به معادله (۲) به‌صورت زیر تعریف کنیم. میانگین پیکسل‌هایی از تصویر B که در فاصله r از پیکسل (x,y) قرار گرفته‌اند. در شکل ۳ این طرح را مشاهده می‌کنید.

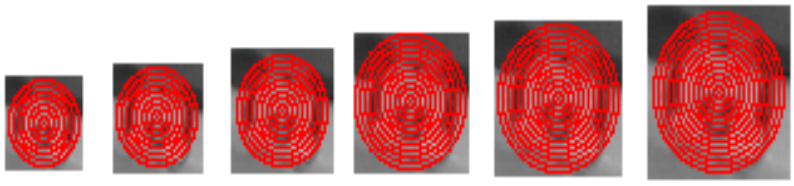
$$C_B(x,y,r)=\frac{1}{2\pi r}\int_0^{2\pi} B(x+r\cos\theta,\ y+r\sin\theta)\,d\theta \qquad (۲)$$

در مرحله اول، الگوی Q به طور مثال با سایز (۵۱*۵۱) را با مقیاس‌های مشخص n=6 تغییر اندازه داده، سپس بر روی الگوهای (Q_1 , Q_2 ,... Q_6) نمونه‌برداری دایره‌ای را با شعاع‌های به فاصله دو پیکسل پیاده‌سازی می‌کنیم.



شکل ۳: طرح‌های دایره‌ای در ماتریسی به ابعاد (۵۱*۵۱)

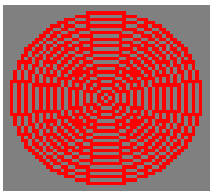
در شکل ۴ الگوهای تغییر سایز یافته با طرح‌های دایره‌ای را مشاهده می‌کنید از روی آن ماتریس CQ که سطرهای آن برابر الگوهای تغییریافته، ستون‌های آن برابر شعاع‌ها و درایه‌های آن برابر میانگین پیکسل‌های واقع‌شده روی هر دایره، تعریف می‌شود. در ماتریس CQ با توجه به سایز الگو Q_i حداکثر شعاع در هر یک از الگوهای تغییر سایز یافته، متفاوت می‌باشد.



شکل ۴: ا الگوهای تغییر سایز یافته با طرح‌های دایره‌ای

در مرحله دوم ابتدا از بین الگوهای تغییر اندازه داده‌شده، اندازه بزرگ‌ترین آن‌ها را برابر (r_1 , c_1) قرار می‌دهیم که در مثال مربوط به Q_6 می‌باشد. سپس با شروع حرکت در سرتاسر پیکسل‌های

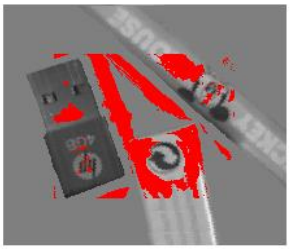
تصویر A به‌اندازه $(\mathbf{r_1}, \boldsymbol{c_1})$، از تصویر A جدا کرده و سپس در مرکز قسمت انتخاب‌شده نمونه‌برداری دایره‌ای را با شعاع‌های به فاصله دو پیکسل پیاده‌سازی می‌کنیم. که در شکل ۵ قسمت انتخاب‌شده از تصویر A با طرح‌های دایره‌ای را مشاهده می‌کنید. از روی آن ماتریس **CA** که دارای یک سطر، ستون‌های آن برابر شعاع‌ها و درایه‌های آن برابر میانگین پیکسل‌های واقع‌شده روی هر دایره، تعریف می‌شود.



شکل ۵: قسمت انتخاب‌شده از تصویر A

در مرحله سوم به‌اندازه طول هر سطر ماتریس CQ از ماتریس CA جدا کرده و بین سطر انتخابی از CQ و CA جدا شده با توجه به معادله ۳، کرولیشن انجام می‌دهیم:

$$corr(x) = \frac{\sum_t \left(CA(t) - \overline{\mathrm{CA}}\right)\left(CQ(t) - \overline{\mathrm{CQ}}\right)}{\sqrt{\sum_t \left(CA(t) - \overline{CA}\right)^2 \sum_t \left(CQ(t) - \overline{CQ}\right)^2}} \qquad (۳)$$



شکل ۶: خروجی فیلترمقیاس

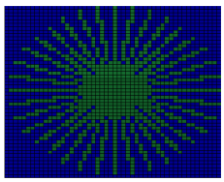
برای تمام سطرهای ماتریسCQ این کار را انجام می‌دهیم. از بین کرولیشن‌های انجام‌شده مقدار ماکزیمم را انتخاب کرده و با معین کردن یک مقدار آستانه، به طور مثال $t_1 = 0.75$، اگر مقدار ماکزیمم کرولیشن‌های انجام‌شده از t_1 بیشتر بود آن پیکسلی از تصویر A که دایره‌ها را به مرکز آن زده‌ایم به‌عنوان پیکسل کاندید نوع اول انتخاب می‌کنیم. در غیر این صورت آن پیکسل را کنار می‌گذاریم. اگر پیکسل به‌عنوان کاندید نوع اول انتخاب شد بایستی توجه کنیم که از کدام سطر ماتریس CQ این به‌دست‌آمده است از روی شماره سطر احتمال مقیاس را به دست می‌آوریم. احتمال مقیاس را بایستی برای آن پیکسل کاندید نوع اول ذخیره کنیم. با حرکت در سرتاسر تصویر A مراحل ۲ و۳ را برای تک‌تک پیکسل‌ها انجام می‌دهیم. همان‌طور که انتظار داشتیم در پایان این فیلتر از بین پیکسل‌های تصویر A، پیکسل‌های کاندید نوع اول با احتمال مقیاس مربوطه

را به دست آوردیم. شکل ۶ نمونه‌ای از پیاده‌سازی فیلتر مقیاس را نشان می‌دهد که پیکسل‌های کاندید نوع اول با رنگ قرمز مشخص شده است.

۳-۲ فیلتر زاویه

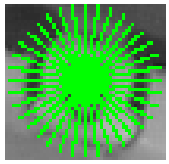
در این فیلتر از یک سری طرح‌های شعاعی روی الگوی Q و تصویر A استفاده می‌کنیم تا در انتهای این فیلتر از بین پیکسل‌های کاندید نوع اول، پیکسل‌های کاندید نوع دوم با احتمال زاویه چرخش مربوطه را به دست آوریم. قبل از شروع پیاده‌سازی این فیلتر لازم است، نمونه‌برداری شعاعی $R_B(x,y,\alpha)$ را با توجه به معادله ۴ به این صورت تعریف کنیم. میانگین پیکسل‌هایی از تصویر B که در روی خط شعاعی به طول L با یک رأس در (x,y) و تحت زاویه α نسبت به افق قرار گرفته‌اند **[۱۰]**. که در شکل ۷ نمونه‌ای از این طرح را مشاهده می‌کنید.

$$R_B^L(x,y,\alpha)=\frac{1}{L}\int_0^L B(x+t\cos\alpha,\ y+t\sin\alpha)\,dt \qquad (۴)۴$$



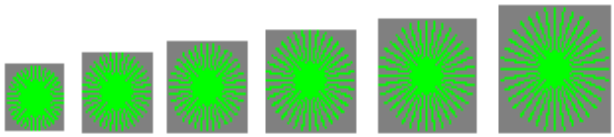
شکل ۷: طرح‌های شعاعی در ماتریسی به ابعاد (۵۱*۵۱)

در مرحله اول این فیلتر، بر روی الگوی Q به طور مثال با سایز (۵۱ *۵۱) ، نمونه‌برداری شعاعی را با زاویه‌های به فاصله ۱۰ درجه و طول خط شعاعی برابر با حداکثر شعاع دایره زده در فیلتر مقیاس (**L=24**) پیاده‌سازی می‌کنیم. در شکل ۸ الگو را با طرح‌های شعاعی مشاهده می‌کنید که از روی آن ماتریس RQ که دارای یک سطر، ستون‌های آن برابر زاویه‌ها و درایه‌های آن برابر میانگین پیکسل‌های واقع‌شده روی خط شعاعی به طول L و زاویه مربوطه، تعریف می‌شود.



شکل ۸: الگو با طرح‌های شعاعی

مرحله دوم بر روی پیکسل‌های تصویر **A** که در فیلتر مقیاس به‌عنوان پیکسل‌های کاندید نوع اول انتخاب شده‌اند انجام می‌گیرد. با توجه به احتمال مقیاس مربوط به پیکسل کاندید نوع اول ، آن را در اندازه الگوی **Q** ضرب می‌کنیم و اندازه به‌دست‌آمده را ($\mathbf{r_2}$, $\boldsymbol{c_2}$) می‌نامیم. همچنین طول خط شعاعی الگو L را نیز در احتمال مقیاس ضرب کرده و $\widehat{\boldsymbol{L}}$ در نظر می‌گیریم. از تصویر A به اندازه ($\mathbf{r_2}$, $\boldsymbol{c_2}$) به مرکزیت پیکسل کاندید نوع اول جدا کرده و سپس روی قسمت انتخاب‌شده، نمونه‌برداری شعاعی را با زاویه‌های به فاصله ۱۰درجه و طول خط شعاعی برابر با $\widehat{\boldsymbol{L}}$ پیاده‌سازی می‌کنیم. با توجه به‌احتمال مقیاس یکی از طرح‌های شکل ۹ را بایستی بر روی قسمت جداشده از تصویر A پیاده‌سازی کنیم.

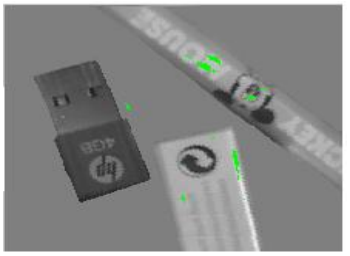


شکل ۹: طرح‌های شعاعی بر روی قسمت جداشده از تصویرA

بر روی طرح انتخاب‌شده از شکل ۹ ماتریس RA که دارای یک سطر، تعداد ستون‌های آن برابر تعداد شعاع‌های رسم شده با زاویه مربوطه بر روی قسمت جداشده از تصویر A و درایه‌های آن برابر میانگین پیکسل‌های واقع‌شده روی خط شعاعی به طول $\widehat{\boldsymbol{L}}$ و زاویه مربوطه، تعریف می‌شود.

در مرحله سوم ابتدا ماتریس RQ را یک شیفت دایره‌ای می‌دهیم. بین ماتریس RA و RQ شیفت داده‌شده با توجه به معادله (۳) ، کرولیشن انجام می‌دهیم. عملیات شیفت **RQ** و محاسبه کرولیشن را به‌اندازه طول ماتریس **RQ** تکرار می‌کنیم. از بین کرولیشن‌های انجام‌شده مقدار ماکزیمم را انتخاب می‌کنیم. با معین کردن یک مقدار آستانه به طور مثال $\mathbf{t_2 = 0.75}$، اگر مقدار ماکزیمم از $\boldsymbol{t_2}$ بیشتر بود آن پیکسل کاندید نوع اول از تصویر **A**، که خط‌های شعاعی را به مرکز آن زده‌ایم به‌عنوان پیکسل کاندید نوع دوم انتخاب می‌کنیم. در غیر این صورت آن پیکسل کاندید نوع اول را کنار می‌گذاریم. اگر پیکسل به‌عنوان کاندید نوع دوم انتخاب شد بایستی توجه کنیم که از چندمین شیفت دایره‌ای ماتریس RQ این به‌دست‌آمده است از روی شماره شیفت دایره‌ای احتمال زاویه چرخش را بایستی برای آن پیکسل کاندید نوع دوم ذخیره کنیم.

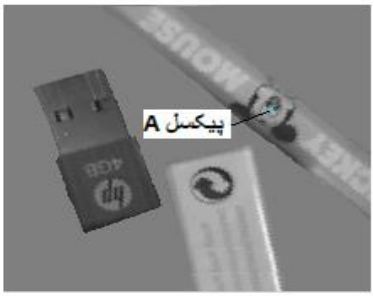
با حرکت در سرتاسر تصویر A مراحل ۲ و۳ را برای تک‌تک پیکسل‌های کاندید نوع اول انجام می‌دهیم. همان‌طور که انتظار داشتیم در پایان این فیلتر از بین پیکسل‌های کاندید نوع اول تصویر A، پیکسل‌های کاندید نوع دوم با احتمال زاویه چرخش مربوطه را به دست آوردیم. شکل ۱۰ نمونه‌ای از پیاده‌سازی فیلترمقیاس را نشان می‌دهد. پیکسل‌های کاندید نوع دوم با رنگ سبز مشخص شده است.



شکل ۱۰: خروجی فیلتر زاویه

۳-۳ فیلتر تطبیق الگو

این فیلتر بر روی پیکسل‌های تصویر $\mathbf{A}$ که در فیلتر زاویه به‌عنوان پیکسل‌های کاندید نوع دوم انتخاب شده‌اند انجام می‌گیرد. همچنین از توضیحات بالا می‌دانیم که، پیکسل‌های کاندید نوع دوم نیز از بین پیکسل‌های کاندید نوع اول انتخاب شده‌اند. پس آن پیکسل کاندید نوع دوم که می‌خواهیم فیلتر تطبیق الگو را برای آن پیاده‌سازی کنیم دارای یک احتمال مقیاس از فیلتر مقیاس و یک احتمال زاویه چرخش از فیلتر زاویه می‌باشد. در مرحله اول، ابتدا الگوی $\mathbf{Q}$ را با توجه به احتمال مقیاس به‌دست‌آمده برای آن پیکسل کاندید نوع دوم، تغییر اندازه داده و سپس با توجه به‌احتمال زاویه چرخش به‌دست‌آمده برای آن پیکسل کاندید نوع دوم می‌چرخانیم. الگو حاصله را $\mathbf{Q}'$ و اندازه آن را $(\mathbf{r_3}\ ,\ \boldsymbol{c_3})$ می‌نامیم. که در شکل ۱ تعدادی از الگوهای تغییر مقیاس و تغییر زاویه یافته را مشاهده می‌کنید. در مرحله دوم، از تصویر $\mathbf{A}$ به اندازه $(\mathbf{r_3}\ ,\ \boldsymbol{c_3})$ به مرکزیت پیکسل کاندید نوع دوم جدا کرده و تصویر جداشده را $\mathbf{A}'$ می‌نامیم. سپس بین الگو $\mathbf{Q}'$ و تصویر $\mathbf{A}'$ با توجه به معادله ۱ کرولیشن انجام می‌دهیم. با معین کردن یک مقدار آستانه به طور مثال $\boldsymbol{t_3} = \mathbf{0.9}$، اگر مقدار $\mathbf{corr}$ از $\boldsymbol{t_3}$ بیشتر بود این پیکسل کاندید نوع دوم به‌عنوان پیکسل کاندید نوع سوم در نظر می‌گیریم. مراحل ۱ و۲ را برای کلیه پیکسل‌های کاندید نوع دوم انجام می‌دهیم. درنهایت مقدار ماکزیمم پیکسل‌های کاندید نوع سوم به‌عنوان نقطه تطبیق الگو به حساب می‌آید. شکل ۱۱ نمونه‌ای از پیاده‌سازی فیلتر تطبیق الگو را نشان می‌دهد. نقطه تطبیق الگو با پیکسل $\mathbf{A}$ به رنگ آبی مشخص شده است.



شکل ۱۱: خروجی فیلتر تطبیق الگو

در شکل ۱۲ برای تصویر هوایی گرفته‌شده، فرآیند تطبیق الگو که شامل ۳ فیلتر متوالی مقیاس، زاویه و تطبیق الگو می‌باشد، پیاده‌سازی شده است. نقطه تطبیق الگو با پیکسل **A** به رنگ سبز مشخص‌شده است.

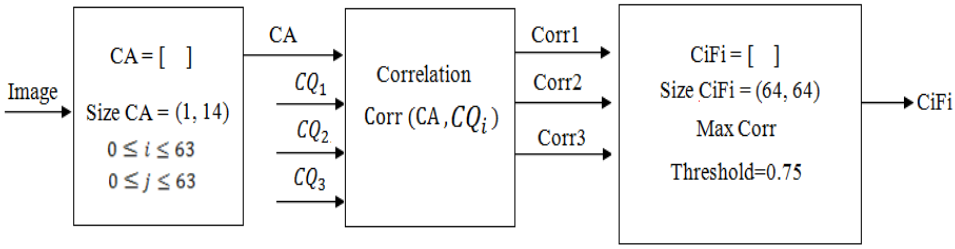
شکل ۱۲: الف) الگو ب) خروجی فیلترمقیاس ج) خروجی فیلتر زاویه د) خروجی فیلتر تطبیق الگو

۴. پیاده‌سازی الگوریتم بهینه بر روی FPGA

در این قسمت نحوه پیاده‌سازی سخت‌افزاری الگوریتم بهینه بر روی الگوی (۵۱ *۵۱) و تصویر (۱۲۰ * ۱۲۰) در ۳ مقیاس (s_1=0.9, s_2=1.0, s_3=1.1) و ۹ زاویه (α_1= -20, α_2= -15, … α_9=20) نشان داده شده است.

۴-۱ پیاده‌سازی فیلتر مقیاس

پیاده‌سازی فیلتر مقیاس در FPGA را می‌توان به سه بلوک اصلی تقسیم کرد. الف- تشکیل ماتریسCA ب- کرولیشن بین ماتریس CAو CQ ج-تشکیل ماتریس CiFi، که در شکل ۱۳ مشاهده می‌کنید.



شکل۱۳: نمایش بلوکی فیلتر مقیاس در FPGA

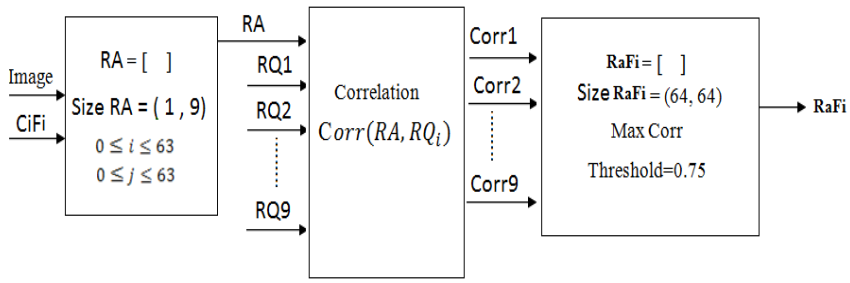
الف- تشکیل ماتریس CA: پیکسل‌های مربوط به طرح‌های دایره‌ای به‌صورت پشت سر هم در داخل ارائه قرار می‌دهیم و متناسب با آدرس ارسالی، مقادیر از تصویر ذخیره‌شده بلوک حافظه فراخوانی می‌شود. چون این اعداد پشت سر هم در داخل ارائه قرار می‌گیرند باید با توجه به تعداد پیکسل‌های هر دایره آن‌ها را از هم تفکیک کرد و میانگین مقادیر پیکسل‌های هر دایره را به ترتیب در داخل ارائه CA قرار داد.

ب-کرولیشن بین ماتریس CA و CQ: پس از تشکیل ماتریس CA از بلوک قبل و ماتریس CQ از الگو با مقیاس‌های مشخص، بایستی بین این دو کرولیشن انجام دهیم. با توجه به مقیاس‌های موجود طول ماتریس CQ در هر حالت متفاوت می‌باشد. به‌اندازه ماتریس CQ از ماتریس CA جدا کرده و بین این دو کرولیشن انجام می‌دهیم. خروجی مربوطه را به ترتیب Corr3,Corr2,Corr1 قرار می‌دهیم.

ج-تشکیل ماتریس CiFi: با توجه به کرولیشن‌های به‌دست‌آمده در بلوک قبل باید ماکزیمم آن‌ها را به دست آورده و مقیاس مربوط به ماکزیمم مقدار را نیز ذخیره کنیم. پس از تعیین مقدار ماکزیمم باید آن را با مقدار آستانه به طور نمونه (t=0.75) مقایسه کنیم که اگر از آن بزرگ‌تر باشد متناسب با مقدار مقیاس عدد مربوطه در CiFi(i,j) قرار می‌دهیم و به‌عنوان پیکسل کاندید نوع اول شناخته می‌شود و اگر کوچک‌تر از مقدار آستانه باشد مقدار صفر را در CiFi(i,j) قرار می‌دهیم.

۲-۴ پیاده‌سازی فیلتر زاویه

پیاده‌سازی فیلتر زاویه در FPGA را می‌توان به سه بلوک اصلی تقسیم کرد. الف- تشکیل ماتریسRA ب- کرولیشن بین ماتریس RAو RQ ج-تشکیل ماتریس RaFi، که در شکل ۱۴ مشاهده می‌کنید.



شکل۱۴: نمایش بلوکی فیلتر زاویه در FPGA

الف- تشکیل ماتریس RA: همان‌طور که در بخش قبل گفته شد فیلتر دوم فقط بر روی پیکسل‌های کاندید نوع اول انجام می‌شود. پس در شروع این فیلتر مقدار CiFi (i, j) چک می‌شود اگر مقدار غیر صفر داشت فیلتر دوم انجام می‌شود در غیر این صورت یک پیکسل جلو رفته و فیلتر اول را انجام می‌دهیم و مقادیر RaFi (i, j)، TiFi (i, j) را برابر صفر قرار می‌دهیم. به این معنا که این پیکسل کاندید نوع اول نمی‌باشد و فیلتر دوم و سوم بر روی آن انجام نمی‌گیرد. پیکسل‌های مربوط به طرح‌های شعاعی به‌صورت پشت سر هم در داخل یکی از سه ارائه Rad1, Rad2, Rad3 قرار گرفته است. با توجه به مقدار CiFi (i, j) که متناسب با مقیاس می‌باشد یکی را انتخاب می‌کنیم و متناسب با آدرس ارسالی مقادیر از تصویر ذخیره‌شده در بلوک حافظه فراخوانی می‌شود. چون این اعداد پشت سر هم در داخل ارائه Rad قرار می‌گیرند باید با توجه به تعداد پیکسل‌های هر خط شعاعی آن‌ها را از هم تفکیک کرده و مجموع مقادیر پیکسل‌های هر خط شعاعی به ترتیب در داخل ارائه RA قرار می‌گیرد. همانند تشکیل CA در فیلتر اول ماتریس RA را نیز تشکیل می‌دهیم.

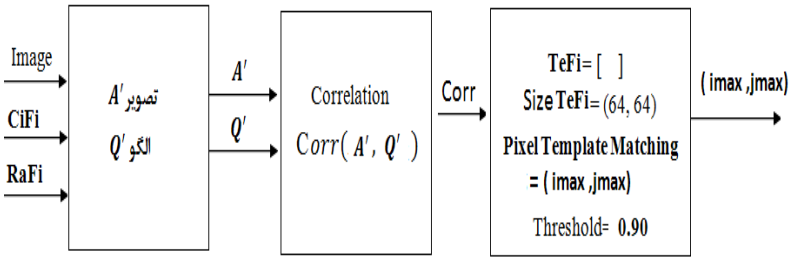
ب-کرولیشن بین ماتریس RA و RQ: در این فیلتر برای انجام کرولیشن بین RA و RQ بایستی ماتریس RQ را با توجه به تعداد زاویه‌ها که در اینجا ۹ عدد می‌باشد شیفت دایره‌ای داده و سپس بین تک‌تک ماتریس‌های ایجادشده و RA کرولیشن انجام دهیم. در ضمن در این قسمت طول دو ماتریس RA و RQ شیفت داده‌شده برابر است. خروجی مربوطه را به ترتیب Corr9, …..Corr2, Corr1 قرار می‌دهیم.

ج-تشکیل ماتریس RaFi: با توجه به کرولیشن‌های به‌دست‌آمده در بلوک قبل باید ماکزیمم آن‌ها را به دست آورده و مقدار زاویه مربوط به ماکزیمم را نیز ذخیره کنیم. پس از تعیین مقدار ماکزیمم باید آن را با مقدار آستانه به طور نمونه (t=0.75) مقایسه کنیم که اگر از آن بزرگ‌تر باشد و متناسب با مقدار زاویه، عدد مربوطه در RaFi(i,j) قرار می‌گیرد و به‌عنوان پیکسل کاندید نوع دوم شناخته می‌شود و اگر از مقدار آستانه کوچک‌تر باشد مقدار ۱۰را در RaFi(i,j) قرار می‌دهیم به

این معنا که این پیکسل کاندید نوع اول هست ولی کاندید نوع دوم نیست و فیلتر سوم بر روی آن انجام نمی‌دهیم و مقدار TeFi(i,j) نیز برابر صفر قرار می‌گیرد. در این صورت یک پیکسل جلو رفته و فیلتر اول را انجام می‌دهیم.

۳-۴ پیاده‌سازی فیلتر تطبیق الگو

پیاده‌سازی فیلتر تطبیق الگو در FPGA را می‌توان به سه بلوک اصلی تقسیم کرد. الف- انتخاب تصویر $\mathbf{A}'$ و الگوی $\mathbf{Q}'$ ب- کرولیشن بین تصویر $\mathbf{A}'$ و الگوی $\mathbf{Q}'$ ج- تشکیل ماتریس TeFi و به دست آوردن نقطه تطبیق الگو، که در شکل ۱۵ مشاهده می‌کنید.



شکل۱۵: نمایش بلوکی فیلتر تطبیق الگو در FPGA

الف-انتخاب تصویر $\mathbf{A}'$ و الگوی $\mathbf{Q}'$: در شروع این فیلتر مقدار CiFi (i, j) و RaFi (i, j) چک می‌شود. چون پیکسل نوع دوم می‌باشد پسCiFi یکی از سه مقدار ۳،۲،۱ را دارد (پیکسل کاندید نوع اول) و به دنبال آن RaFi نیز باید مقدار غیر ۱۰و عددی بین ۱ تا ۹ را داشته باشد (پیکسل کاندید نوع دوم). در غیر این‌صورت یک پیکسل جلو رفته و فیلتر اول را انجام می‌دهیم و TeFi =0 (i, j) قرار می‌دهیم. از مقیاس ذخیره‌شده در CiFi (i, j) و زاویه ذخیره‌شده در RaFi(i,j) استفاده کرده و متناسب با آن‌ها الگوی تغییر اندازه یافته و چرخیده شده را از بلوک‌های حافظه فراخوانی می‌کنیم و آن را الگوی $\mathbf{Q}'$ می‌نامیم. متناسب با اندازه الگوی به‌دست‌آمده در مرحله قبل یکی از آرایه‌های Tem3,Tem2,Tem1 را استفاده کرده و متناسب با آدرس ارسالی مقادیر از تصویر ذخیره‌شده بلوک حافظه فراخوانی می‌شود و آن را تصویر $\mathbf{A}'$ قرار می‌دهیم.

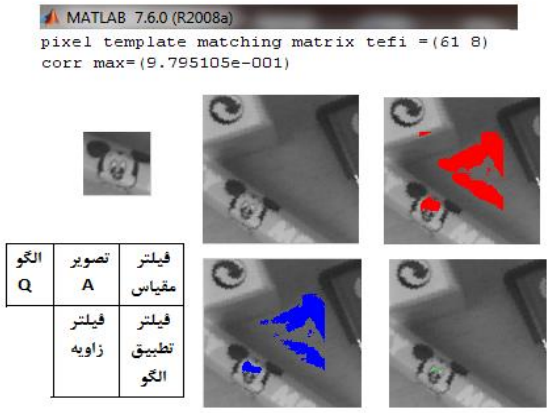
ب-کرولیشن بین تصویر $\mathbf{A}'$ و الگوی $\mathbf{Q}'$: پیاده‌سازی کرولیشن بین تصویر $\mathbf{A}'$ و الگو $\mathbf{Q}'$ انجام می‌دهیم و خروجی مربوطه را برابر Corr قرار می‌دهیم.

ج-تشکیل ماتریس TeFi و به دست آوردن نقطه تطبیق الگو: در این قسمت مقدار کرولیشن ماکزیمم را در ابتدا برابر مقدار آستانه این فیلتر که به طور نمونه (t=0.90) می‌باشد قرار می‌دهیم. سپس مقدار corr به‌دست‌آمده را با مقدار کرولیشن ماکزیمم مقایسه می‌کنیم اگر بزرگ‌تر بود آن را برابر کرولیشن ماکزیمم قرار داده و همچنین مقدار i و j را نیز برابر imax و jmax قرار می‌دهیم

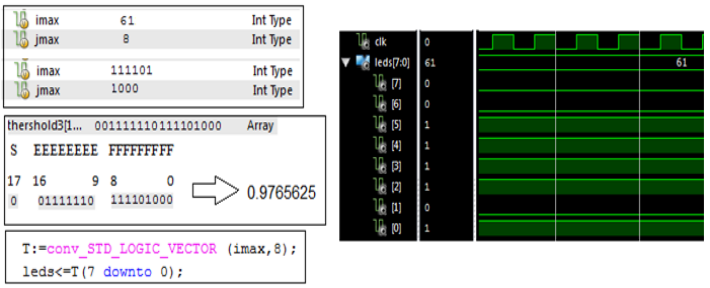
و اگر کوچک‌تر بود تغییر در مقدار کرولیشن ماکزیمم رخ نمی‌دهد. در هر دو حالت مقدار TeFiرا برابر ۱۱ قرار می‌دهیم به این معناست که پیکسل مربوطه هم کاندید نوع اول و هم نوع دوم است. وقتی‌که تمام عملیات مربوط به هر سه فیلتر برای تمام تصویر انجام شد در این مرحله imax, jmax مختصات نقطه تطبیق الگو را نشان می‌دهد که در ماتریس TeFi با عدد ۱۵ نشان می‌دهیم. در ضمن بعد از انجام عملیات برای یک تصویر بایستی تعدادی از متغیرها ازجمله corrmax را به حالت اولیه برگرداند.

۵. نتایج پیاده‌سازی بر روی FPGA

پس از پیاده‌سازی سخت‌افزاری الگوریتم انتخاب‌شده بر روی FPGA با استفاده از تراشه، XC3SD3400A از خانواده Spartan-3A DSP، به عمل تطبیق الگوی (۵۱ *۵۱) در ۳ مقیاس (s_1=0.9, s_2=1.0, s_3=1.1) و ۹ زاویه (α_1= -20, α_2= -15, … α_9=20) در مدت‌زمان حدود ۰/۱ ثانیه بر روی تصویر (۱۲۰ * ۱۲۰) دست می‌یابیم. در شکل‌های ۱۶ و ۱۷ با یک مثال خروجی‌های به‌دست‌آمده در برنامه MATLAB و برنامه VHDL نشان داده شده است. در شکل ۱۸ برد FPGA مورداستفاده و تصویری از پیاده‌سازی و شبیه‌سازی الگوریتم بهینه تطبیق الگو نشان داده شده است



شکل۱۶: مجموعه جواب‌های به‌دست‌آمده از شبیه‌سازی در MATLAB



شکل۱۷: مجموعه جواب‌های به‌دست‌آمده از پیاده‌سازی در FPGA

شکل۱۸: تصویری از برد FPGA مورداستفاده و نمایی از پیاده‌سازی و شبیه‌سازی الگوریتم بهینه

۶. نتیجه‌گیری

در این مقاله الگوریتمی بهینه از تطبیق الگو مبتنی بر ضرایب کرولیشن، تغییرناپذیر نسبت به مقیاس و زاویه بیان شد. در یک مقایسه انجام‌شده بین الگوریتم بهینه‌شده و الگوریتم brute force با استفاده از یک کامپیوتر Pentium4-3GHz تصویر A به‌اندازه (۴۶۵*۳۳۸)، الگو به‌اندازه (۵۱*۵۱)، مقیاس با گام ۰/۱ (s_1=0.6, s_2=0.7, ….s_6=1.1) و زاویه با گام ۱۰ درجه (α_1=0, α_2=10,...α_{36}=350)، با الگوریتم brute force در حدود ۹۲۰۰ ثانیه طول می‌کشد. درحالی‌که الگوریتم بهینه‌شده آن را در ۲۲ ثانیه انجام می‌دهد که حدود ۴۰۰ بار سریع‌تر از آن می‌باشد. درحالی‌که نتایج یکسانی از هر دو الگوریتم به دست می‌آید. الگوریتم بهینه‌شده به دلیل استفاده مستقیم از تصاویر نسبت به الگوریتم‌هایی که ساده‌سازی قبلی روی تصویر انجام می‌دهند دارای دقت بیشتری می‌باشد. همچنین پس از پیاده‌سازی سخت‌افزاری الگوریتم انتخابی بر روی FPGA با استفاده از XC3SD3400A از خانواده Spartan-3A DSP عمل تطبیق الگوی (۵۱ * ۵۱) در ۳ مقیاس (s_1=0.9, s_2=1.0, s_3=1.1) و ۹ زاویه (α_1= -20, α_2= -15, … α_9=20) در مدت‌زمان حدود ۰/۱ ثانیه بر روی تصویر (۱۲۰ * ۱۲۰) انجام پذیرفت. به‌عبارت‌دیگر تطبیق الگو تغییرناپذیر به مقیاس و چرخش، را در حدود ۱۰ فریم در ثانیه به انجام رساندیم.

۷. قدردانی

با تشکر از دانشکده مهندسی برق دانشگاه، که امکانات آزمایشگاهی را در اختیار اینجانب قرار دادند.

۸. منابع

[1] N. Mayasari and A. P. Siahaan, “Vehicle Plate Recognition using Template Matching” International Journal For Innovative Research In Multidisciplinary Field , Vol 4, pp.259-263, 2018.

[2] S. Caraiman et al., “Computer Vision for the Visually Impaired: the Sound of Vision System,” in 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 1480–1489,2017.

[3] S. Nikan and M. Ahmadi, “Partial Face Recognition Based on Template Matching,” in 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), pp. 160–163. 2015.

[4] F. Essannouni “Fast Frequency Template Matching Using Higher Order Statistics,” IEEE Trans. Image Process., Vol. 19, No. 3, pp. 826–830, 2010.

[5] M. Suzuki, Y. Tanida, T. Maruyama, “Detecting patterns in various size and angle using FPGA”, Int. Conf. on Field Programmable Logic and Applications, pp.151-154, 2010.

[6] W. Chia Lee, C. Hsing Chen, “A Fast Template Matching Method Rotation Invariance Using Two Stage Process” , Int. Conf. on Intelligent Information Hiding and Multimedias Signal Processing, pp.109-112, 2009.

[7] L. Ming, j.Z. Guang, “A novel algorithm for a rotation invariant template matching”, Optoelectron. Lett, Vol. 4, No 5, pp. 379-383, 2008.

[8] J.Flusser, T.Suk, “Rotation moment invariant for recognition of symmetric objects” IEEE T.Image Processing, Vol. 15, No 12, pp.3787-3790, 2006.

[9] C.R.P. Dionisio, H.Y. Kim, “A supervised shape classification technique invariant under rotation and scaling”, Int. Telecommunications Symposium, pp.533-537,2002.

[10] H.Y. kim, S.A. Araujo, “grayscale Template Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast” IEEE pacific-Rim Symposium on Image and Video Technology. Leacture Notes in Computer Science, Vol. 4872, pp.100-113, 2007.

[11] Y. Tanida, T. Maruyama, “An Approach for Downscaling Images for Real-time Pattern Detection”, pp.253- 256 2008.

[12] M. Arif “Correlation-Coefficient-Based Fast Template Matching Through Partial Elimination” IEEE Signal Processing Society. Vol. 21, No. 4, 2012.

[13] A. Lindose, L. Entrena, “High performance FPGA-based image correlation”, J.Real-time Image Proc, pp.223-233, 2007.

[14] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Englewood Cliffs, NJ: Prentice Hall, 2014.