

پیاده‌سازی یک ربات متحرک بینا با استفاده از یک الگوریتم شناسایی رنگ جدید و الگوی هدایت ترکیبی

مهدي سعادت‌مند

دانشجوی دکتری مهندسی پزشکی-بیوالکترونیک، دانشگاه تربیت مدرس
saadatmand@kiaeee.org

مهدي ملبوبي

کارشناس ارشد، مهندسی مخابرات-سیستم، دانشگاه فردوسی مشهد
mehdi_malboubi@yahoo.com

مرتضی خادمی

عضو هیات علمی دانشگاه فردوسی مشهد
khademi@um.ac.ir

محمد رضا اکبرزاده توتونچی

عضو هیات علمی دانشگاه فردوسی مشهد
akbarzadeh@ieee.org

چکیده

روش‌هایی که تاکنون برای هدایت ربات‌ها ارائه شده است را می‌توان به سه دسته کلی سلسله مراتبی، رفتارگرا و ترکیبی تقسیم نمود. در این مقاله، یک ربات متحرک بینا با معماری ترکیبی شامل پنج رفتار واکنشی و سه رفتار هدف‌گرا ارائه شده است. کلیه رفتارهای هدف‌گرای ربات به همراه رفتار واکنشی جستجوی تصویری هدف، از اطلاعات سطح بالای حاصل از پردازش تصویر استفاده می‌کنند. در حالی که سایر رفتارهای واکنشی برپایه اطلاعات حاصل از حسگرهای جانبی ربات شامل حسگرهای مادون قرمز (برای تشخیص خط هادی) و حسگرهای لامسه بازو طرح‌ریزی شده‌اند. بعلاوه، هر سه رفتار هدف‌گرای ربات با استفاده از یک کنترل کننده تناسبی-مشتق‌گیر-تناسبی با پسخورد تصویری پیاده‌سازی گردیده است. برای شناسایی اشیاء از ویژگی رنگ آنها استفاده شده است. بر این اساس، در این مقاله یک الگوریتم جدید با عنوان فیلتر منطبق رنگ برای شناسایی رنگ ارائه شده است. این الگوریتم از دو مرحله طراحی و بازرسی تشکیل شده است. در مرحله طراحی از الگوریتم‌های ژنتیکی برای بهینه‌سازی ضرایب فیلتر استفاده می‌گردد. با اینکه روش تکاملی فوق، یک الگوریتم خارج خط است، اما الگوریتم بازرسی پیشنهادی از حجم محاسباتی کمی برخوردار بوده و دارای قابلیت پیاده‌سازی به صورت بی‌درنگ است. نتایج تجربی بیانگر عملکرد قابل قبول فیلترهای منطبق در شناسایی سریع رنگ تحت شرایط نوری متغیر است. کنترل کننده ربات نیز قابلیت اطمینان چشم‌گیری به ربات بخشیده است. به طوری که در آزمایشات تجربی، هیچ‌گاه مشاهده نگردید که ربات در حرکت به سوی هدف از پیش تعیین شده، دچار خطا گردد. هدایت کننده نیز عملکرد خوبی از خود به نمایش گذارده است. طی آزمایشات تجربی، ربات به آسانی توانسته است بارها و بارها همه اشیاء موجود در زمین مسابقه را شکار و در دروازه مربوطه بیاندازد.

کلمات کلیدی

ربات متحرک، معماری هدایت ترکیبی، کنترل کننده دیداری، تشخیص رنگ، الگوریتم‌های ژنتیکی

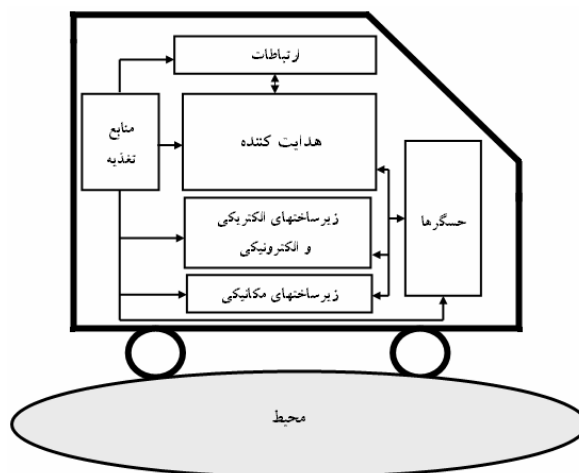
۱) مقدمه

یک ربات هوشمند، ماشینی خودکار و چند منظوره، با قابلیت انجام مجموعه‌ای از وظایف (همانند انسان)، تحت شرایط مختلف است. اینگونه ربات‌ها قادر به درک و تعامل با محیط اطراف خود هستند [۱]. دستاوردهای جدید بشر در زمینه‌های گوناگون، دانش رباتیک را وارد مرحله جدیدی از حیات خویش کرده است. هدف نهایی دانش رباتیک در هزاره جدید، خلق رباتی است که همانند انسان خصوصیات برجسته‌ای در رفتار، حرکت، هوش و ارتباط از خود به نمایش بگذارد [۲].

ربات متحرک، ماشین هوشمندی با قابلیت جابجایی از یک نقطه به نقطه دیگر است. این ماشین قادر است با استفاده از اطلاعات دریافتی از حسگرهای خود، محیط اطراف را شناسایی کرده و بنا به شرایط، از خود واکنش‌های مناسب نشان دهد [۳]. دو ویژگی قابلیت جابجایی و تعامل با محیط باعث توسعه حوزه کاربرد ربات‌های متحرک شده است. شکل ۱ واحدهای مختلف تشکیل دهنده یک ربات متحرک را نشان می‌دهد [۴]. در میان بخش‌های مختلف تشکیل دهنده یک ربات، هدایت کننده پراهمیت‌ترین بخش می‌باشد. به طور کلی، روش‌های متداول در هدایت ربات‌ها را می‌توان به سه دسته تقسیم نمود: (۱) الگوهای سلسله مراتبی^۱، (۲) رفتارگرا^۲ و (۳) ترکیبی^۳ [۳-۹].

الگوهای سلسله مراتبی قدیمی‌ترین الگوی هدایت ربات‌ها می‌باشند. در اینگونه روش‌ها، سعی بر این است که در هر لحظه، براساس اطلاعات دریافتی از حسگرها و دانش موجود، بهترین تصمیم اتخاذ و اجرا شود. معمولاً یک هدایت کننده سلسله مراتبی دارای ساختاری لایه-لایه و سلسله مراتبی است. وظیفه لایه‌های پایین‌تر جمع‌آوری اطلاعات حسگرها و اجرای فرآیند است، در حالی که لایه‌های بالاتر بیشتر در امر تصمیم‌گیری و برنامه‌ریزی دخیل هستند. با حرکت از لایه‌های پایینی به سمت لایه‌های بالاتر، داده‌ها معنی‌دارتر شده و از حجم آنها کاسته می‌شود. در این نوع معماری، ربات به دانش کامل و دقیقی از پیرامون خود نیاز دارد. به همین جهت، معمولاً الگوهای سلسله مراتبی پیچیده بوده و از حجم محاسباتی زیادی برخوردار هستند [۱]. در حقیقت معماری سلسله مراتبی بر اطلاعات کلی^۴ ربات از محیط اطرافش تأکید دارد.

الگوهای رفتارگرا با الهام از سیستم‌های زیستی پیشنهاد شده‌اند. در این نوع معماری، بر خلاف معماری سلسله مراتبی، به جای مدل‌سازی عدم قطعیت‌های محیط، نگاشتنی از فضای حسگرها به فضای عمل ایجاد می‌گردد. در الگوهای رفتارگرا برای هر حس خاص، یک عمل متقابل وجود دارد. در حقیقت، در این نوع معماری، مدل محیط در میان رفتارها توزیع می‌شود و هر رفتار تنها از قسمتی از مدل محیط (که به آن احتیاج دارد)، استفاده می‌کند. وظیفه مدیریت و ایجاد هماهنگی بین رفتارهای مختلف، برعهده واحد برنامه‌ریزی ربات است.



شکل ۱. واحدهای مختلف تشکیل دهنده یک ربات متحرک و چگونگی ارتباط آنها.

^۱ Hierarchical paradigms

^۲ Behavioral paradigms

^۳ Hybrid paradigms

^۴ Global information

با استفاده از مدل‌های رفتارگرا، ربات قادر است با ترکیب دو یا چند رفتار ساده پایه، عملکرد پیچیده‌ای از خود نشان دهد [۹]. الگوهای رفتارگرا بر اطلاعات محلی^۱ ربات از محیط اطرافش تأکید می‌ورزند. رفتارها را می‌توان به دو دسته واکنشی و هدف‌گرا تقسیم نمود. آن دسته از رفتارها که بر اساس وقوع رویدادهای خاصی طراحی شده‌اند، از نوع واکنشی و آن دسته که ربات را در جهت دستیابی به هدفی خاص هدایت می‌کنند، از نوع هدف‌گرا هستند. رفتارهای واکنشی معمولاً از اطلاعات سطح پائین‌تر مانند اطلاعات حسگرها، و رفتارهای هدف‌گرا از تفسیر یا توصیف‌های خلاصه شده، مانند توصیف‌های حاصل از الگوریتم‌های پردازش تصویر، که بوسیله سطوح بالاتر ادراکی ساخته می‌شوند، استفاده می‌کنند.

معماری‌های ترکیبی کارآمدترین شیوه هدایت ربات‌ها در محیط‌های غیرساخت‌یافته و پویا هستند. در این نوع معماری، رفتارهای ربات با استفاده از یک ساختار سلسله مراتبی مدیریت و برنامه‌ریزی می‌شود. به این ترتیب، این نوع معماری از مزایای هر دو معماری سلسله مراتبی و رفتارگرا برخوردار است. در معماری ترکیبی، اطلاعات محلی و اطلاعات کلی به شکلی کارآمد با یکدیگر ترکیب می‌شوند. به این ترتیب که در فرآیند هدایت ربات، معمولاً رفتارها از اطلاعات محلی (سطح پایین) و واحد برنامه‌ریزی و مدیریت (با ساختار سلسله مراتبی خود) از اطلاعات کلی (سطح بالا) استفاده می‌کنند [۷-۹].

۱-۱) معرفی چند روش متداول در هدایت ربات‌های متحرک

تاکنون تحقیقات گسترده‌ای بر روی روش‌های هدایت ربات‌های متحرک انجام شده است. به عنوان مثال، McFetridge و Ibrahim [۱۰] با ترکیب منطق فازی و میدان‌های پتانسیل، روشی مقاوم برای تعیین مسیر ربات‌های متحرک به منظور پرهیز از مانع^۲ ارائه داده‌اند. Liu و Luh [۱۱] با ترکیب اطلاعات مبتنی بر رفتار و یک مکانیزم بهینه‌سازی برخط مبتنی بر ویژگی‌های سیستم ایمنی بدن انسان، روشی جدید برای هدایت ربات در فضاهای ناشناخته و پیچیده ارائه داده‌اند. آنها نشان دادند که روش فوق قادر به هدایت ربات در پرهیز از موانع و رسیدن به هدف می‌باشد. Huq و همکارانش [۱۲] از یک معماری ترکیبی برای هدایت ربات استفاده کرده‌اند. در این روش، ابتدا با استفاده از اطلاعات کلی مسیر امنی برای ربات بروش دیگرام Voronoi طراحی می‌شود. سپس، گره‌های این مسیر به عنوان هدف‌های محلی ربات در نظر گرفته شده و توسط رفتارهایی چون پرهیز از موانع و کوتاهترین مسیر دنبال می‌گردد. در کار دیگری، Labrosse [۱۳] از تصاویر با دید وسیع^۳ برای هدایت ربات در مسیری که با نقاط شاخص مشخص گردیده، استفاده نموده است. Hayet و همکارانش [۱۴] الگوریتمی برای تشخیص علامت‌های چهارگوش در محیط‌های داخلی مانند راهروها با استفاده از پردازش تصاویر ارائه داده‌اند. این علائم مربوط به اشیاء مسطحی از قبیل پوستر، در و قفسه در مسیر حرکت ربات می‌باشد. از الگوریتم فوق برای هدایت یک ربات متحرک استفاده شده است؛ به این ترتیب که همزمان با پیش‌روی ربات، دائماً علائم موجود در مسیر جستجو شده و مسیر حرکت ربات بهنگام می‌گردد. این الگوریتم تحت شرایط نوری گوناگون نیز عملکرد مناسبی داشته است. Huang و همکارانش [۱۵] برپایه مدل Warren و Fajen (برای حرکت انسان)، روشی مبتنی بر یک دوربین برای هدایت ربات‌های متحرک ارائه داده‌اند. در این مدل، شتاب زاویه‌ای ربات تنها با استفاده از جهت نسبی و فاصله آن تا هدف و موانع محاسبه می‌شود. اخیراً، Fernandez-Leon و همکارانش [۱۶] به مطالعه افزایش رفتارها در ربات‌های تکاملی پرداخته است. در اینگونه ربات‌ها، از ترکیب تکاملی رفتارهای ساده، رفتارهای پیچیده‌ای بدست می‌آید. در این تحقیق، هر رفتار توسط یک شبکه عصبی مصنوعی ایجاد شده و امکان ترکیب آنها توسط یک ساختار سلسله مراتبی فراهم شده است. نتایج تجربی بیانگر آن است که رفتارهای ترکیبی قادر به حل مسائل جهان واقعی می‌باشند؛ هرچند که نتایج بسیار به دانش اولیه کاربر درباره مسأله وابسته است.

۲-۱) ربات پیشنهادی

در این مقاله یک ربات متحرک بینا برپایه معماری هدایت ترکیبی ارائه شده است. ربات مذکور از مجموعه‌ای از رفتارهای هدف‌گرا و واکنشی برای حل مسأله کلاسیک جمع‌آوری اشیاء استفاده می‌کند. در این ربات از حسگرهای مادون قرمز برای تشخیص خط هادی،

¹ Local information

² Obstacle avoidance

³ Panoramic images

حسگرهای لامسه برای کنترل بازو و دوربین دیجیتالی برای شناسایی محیط استفاده شده است. ربات با پردازش تصاویر دریافتی از دوربین، بر اساس ویژگی‌های رنگی، اشیا موجود در صحنه پیش روی خود را شناسایی می‌کند. برای این منظور، در این مقاله، یک روش تشخیص رنگ جدید، تحت عنوان فیلترهای منطبق رنگ^۱ (CMF) ارائه شده است. الگوریتم پیشنهادی شامل دو مرحله طراحی و بازرسی است. در مرحله طراحی ضرایب فیلتر با استفاده از الگوریتم‌های ژنتیکی بهینه‌سازی گردیده و در مرحله بازرسی از فیلتر بهینه حاصل برای شناسایی اشیا استفاده می‌شود. با اینکه روش تکاملی طراحی از نظر محاسباتی پیچیده است، اما الگوریتم بازرسی از نظر محاسباتی ساده بوده و از کارایی مناسبی برخوردار است. لذا، می‌توان از فیلترهای منطبق رنگ در کاربردهای بی‌درنگ سود جست. در این ربات از سه فیلتر منطبق رنگ قرمز، زرد و سیاه استفاده شده است.

مهمترین رفتار هدف‌گرایی^۲ ربات، تعقیب تصویری هدف^۳ (VGF) است. این رفتار با استفاده از یک کنترل‌کننده تناسبی-مشتق‌گیر-تناسبی^۴ (PD-P) با پاسخورد تصویری، ربات را در جهت نزدیک شدن به هدفی از پیش تعیین شده هدایت می‌کند. با ترکیب VGF و فیلترهای منطبق رنگ، دو رفتار هدف‌گرایی دیگر شکل می‌گیرد: ره‌گیری تصویری خط هادی^۵ (VLF) و شکار تصویری هدف^۶ (VGH). در کنار سه رفتار هدف‌گرایی فوق، پنج رفتار واکنشی دیگر نیز برای ربات در نظر گرفته شده است: (۱) جستجوی حسگری هدف^۷ (IGS) و (۲) ره‌گیری حسگری خط هادی^۸ (ILF) براساس تقابل با حسگرهای مادون قرمز؛ (۳) گرفتن شی^۹ (OG) و (۴) رها کردن شی^{۱۰} (OD) برپایه برهم‌کنش با حسگرهای لامسه بازو؛ و (۵) جستجوی تصویری هدف^{۱۱} (VGS) با استفاده از فیلترهای منطبق رنگ.

نتایج تجربی بیانگر عملکرد بسیار رضایت‌بخش ربات است. ربات با بکارگیری رفتارهای مختلف بخوبی قادر است، اشیا را شناسایی و شکار نموده و به محل‌های از پیش تعیین شده منتقل نماید. همچنین نتایج تجربی نشان می‌دهند که فیلترهای منطبق رنگ در ناحیه‌بندی تصاویر و آشکارسازی اشیا برپایه رنگ بسیار توانمند هستند. به طوری که این فیلترها قادرند تحت شرایط نوری بسیار متنوع، اشیا را بخوبی شناسایی و از تصویر استخراج نمایند.

۱-۳ ساختار مقاله

در ادامه مقاله، در بخش دوم صورت مسأله‌ای که ربات پیشنهادی برای آن طراحی گردیده، شرح داده شده است. در بخش ۳، ساختار ربات به صورت خلاصه معرفی گردیده است. بخش ۴ به بررسی هدایت‌کننده ربات اختصاص یافته است. بخش ۵ هدایت‌کننده دیداری ربات را معرفی می‌کند. بخش ۶ به شرح الگوریتم‌های پردازش تصویر مورد استفاده در ربات، از جمله روش پیشنهادی فیلتر منطبق رنگ اختصاص یافته است. در بخش ۷، نتایج تجربی ارائه گردیده و در نهایت، بخش ۸ به نتیجه‌گیری اختصاص یافته است. نمادهای مورد استفاده در این مقاله، استاندارد و رایج بوده و ماتریس‌ها/مجموعه‌ها با حروف بزرگ و بردارها با حروف بزرگ و پررنگ نشان داده شده‌اند.

۲ توصیف مساله

هر ربات برای هدفی خاص، طراحی و پیاده‌سازی می‌شود. ما ربات مطرح در این مقاله را برای شرکت در سطح دو مسابقات سراسری رباتیک ایران طراحی کرده‌ایم. بر اساس صورت مسأله مطرح در این مسابقات، ربات باید طی کمترین زمان ممکن، اشیا قرار گرفته در زمین مسابقه را شکار کرده و در دروازه مربوطه بیاندازد.

¹ Color Match Filter (CMF)

² Goal directed behavior

³ Visual Goal Following behavior (VGF)

⁴ Proportional-Differential-Proportional controller (PD-P)

⁵ Visual Line Following behavior (VLF)

⁶ Visual Goal Hunting behavior (VGH)

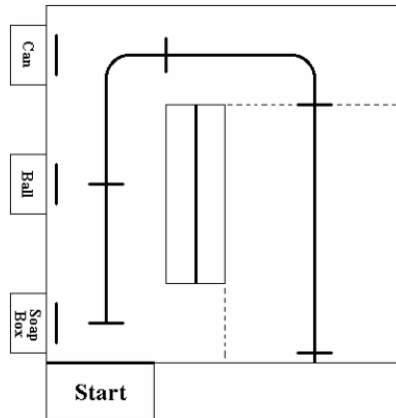
⁷ Infrared-based Goal Seeking behavior (IGS)

⁸ Infrared-based Line Following behavior (ILF)

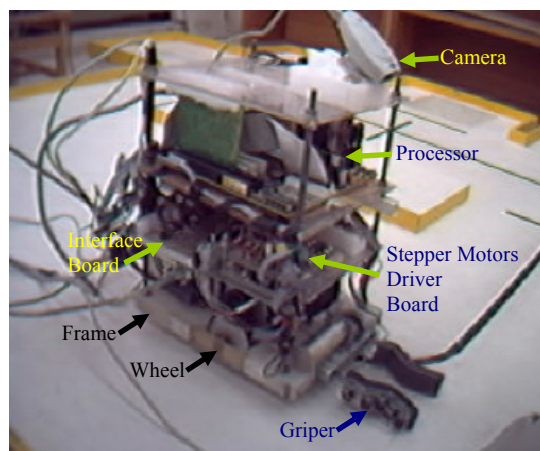
⁹ Object Grasping behavior (OG)

¹⁰ Object Delivering behavior (OD)

¹¹ Visual Goal Seeking behavior (VGS)



شکل ۲. زمین مسابقه



شکل ۳. ربات ساخته شده و بخش‌های مختلف آن.

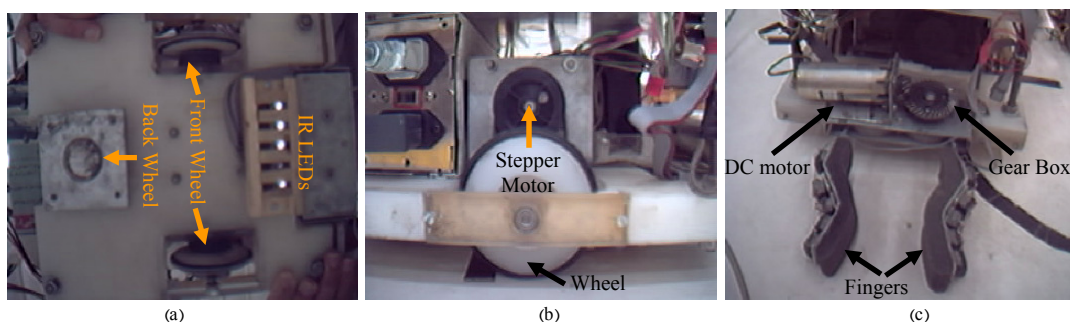
زمین مسابقه در شکل ۲ نشان داده شده است. سه جسم در زمین قرار می‌گیرد: توپ زرد رنگ، جعبه صابون قرمز رنگ و قوطی نوشابه سیاه. زمین مسابقه سفید و خط‌های (خط ضخیم در شکل ۲) سیاه رنگ است. همه اجسام در سمت راست زمین، درون محوطه‌ای که در شکل با خط چین مشخص شده است، قرار می‌گیرند و ربات در آغاز مسابقه در محل شروع گذارده می‌شود.

۲) ساختار ربات طراحی شده

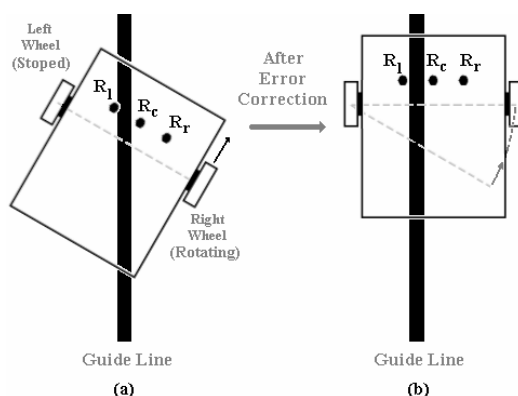
ما به منظور بررسی تئوری‌های ارائه شده در این مقاله و انجام نتایج تجربی، یک ربات متحرک بینا را طراحی و پیاده‌سازی نمودیم. در شکل ۳، ربات ساخته شده و بخش‌های مختلف آن نشان داده شده که در ادامه معرفی می‌گردند.

۱-۳) زیرساخت‌های مکانیکی

زیرساخت‌های مکانیکی ربات متحرک ساخته شده عبارت است از شاسی، سیستم رانش، بازوی دو انگشتی و مکانیزم‌های کمکی مانند تسمه‌های انتقال نیرو. به دلیل مزایایی چون سبکی و تراشکاری آسان، شاسی ربات از جنس پلاستیک فشرده ساخته شده و کلیه اجزا بر روی آن سوار گردیده است. همانطور که در شکل ۴-a نشان داده شده، سیستم رانش ربات از نوع رانش تفاضلی [۸] با دو چرخ متحرک در جلو و یک چرخ هرزگرد در عقب است. در این روش، اختلاف سرعت دو چرخ جلو نقش فرمان را نیز ایفا می‌کند. چرخ عقب ربات نیز از نوع کروی بوده و دارای قابلیت حرکت در تمام جهات است. نیروی محرکه چرخ‌های جلو توسط دو موتور الکتریکی پله‌ای با گام $1/8^\circ$ و توان 6W تأمین می‌شود. برای انتقال نیروی موتور به چرخ از تسمه استفاده شده است (شکل ۴-b را ببینید). مطابق شکل ۴-c، بازوی ربات که در جلوی آن نصب شده است، از نوع دو انگشتی می‌باشد. نیروی محرکه بازو با استفاده از یک موتور DC تأمین گردیده و توسط یک جعبه دنده به بازو منتقل می‌شود.



شکل ۴. برخی از زیرساخت‌های مکانیکی و الکترونیکی ربات از نمای نزدیک. (a) بازوی دو انگشتی، (b) سیستم انتقال نیرو از موتور به چرخ و (c) چرخ‌های جلو و عقب.



شکل ۵. نحوه قرار گرفتن حسگرهای مادون قرمز در زیر ربات و اثر آن در فرآیند تصحیح خطا.

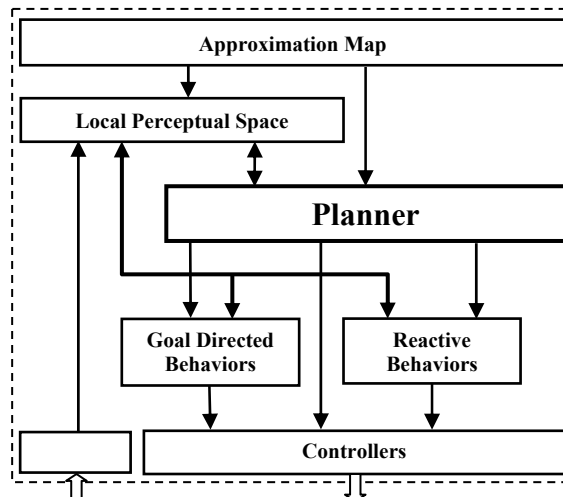
۲-۳) زیرساخت‌های الکترونیکی

زیرساخت‌های الکترونیکی ربات شامل واحد پردازش، مدار واسط^۱، حسگرها و مدار محرک موتورهای پله‌ای^۲ تشکیل شده است (شکل ۳ را ببینید). در این ربات از یک پردازنده Pentium II 333 MHz با حافظه 256 MB برای انجام کلیه پردازش‌ها استفاده شده است. برای تبادل اطلاعات بین پردازنده با حسگرها، کنترل‌کننده‌ها و محرک‌ها از یک مدار واسط شامل یک درگاه ورودی-خروجی و یک شمارنده-زمان‌سنج استفاده شده است. از درگاه ورودی-خروجی برای جمع‌آوری داده‌های حسگرها و تحریک محرک‌ها استفاده گردیده است در حالی که شمارنده-زمان‌سنج نقش منبع تولید پالس برای موتورهای پله‌ای را برعهده دارد. به عبارت دقیقتر، مدار محرک موتورهای پله‌ای با استفاده از سیگنال تولید شده توسط مدار واسط، توان کافی برای تحریک سیم‌پیچ‌های موتورهای پله‌ای را تأمین نموده و سرعت آنها را تنظیم می‌نماید. فرآیند کنترل دور موتورها به طور خلاصه به شرح ذیل است: پردازنده با توجه به اطلاعات دریافتی از محیط، سرعت موتورها را تعیین و بر اساس آن فرکانس کار زمان‌سنج موجود بر روی برد واسط را تنظیم می‌نماید. با تعیین فرکانس کار زمان‌سنج، فرکانس سیگنال تحریک موتورها نیز تغییر کرده و در نتیجه موتورها با سرعت مورد نظر حرکت خواهند کرد.

حسگرهای مورد استفاده در این ربات عبارتند از: (۱) حسگرهای مادون قرمز تشخیص خط هادی، (۲) حسگرهای لامسه (در بازوی ربات) و (۳) دوربین دیجیتالی. همانطور که در شکل ۳ نشان داده شده، دوربین بر روی پیشانی ربات نصب گردیده و دامنه دید آن تقریباً ۴۵ سانتیمتر می‌باشد. ما در این ربات مطابق شکل ۵-a، از سه حسگر مادون قرمز که در راستای عرضی در زیر ربات قرار گرفته‌اند، استفاده کرده‌ایم. راستای قرارگیری حسگرها، کمی جلوتر از راستای چرخ‌های ربات است. همانطور که در شکل ۵-b نشان داده شده است، این موضوع می‌تواند در فرآیند تصحیح خطای حرکت ربات، تأثیر مثبت داشته باشد. خصوصاً اینکه در هنگام تصحیح خطا، یک چرخ ثابت بوده و دیگری با سرعت ثابت می‌چرخد.

¹ Interface board

² Stepper motors driver board



شکل ۶. بخش‌های مختلف هدایت کننده ربات و ارتباط بین آنها.

۴) هدایت کننده ربات

در این ربات، به منظور استفاده از مزایای الگوهای سلسله مراتبی و رفتارگرا از هدایت کننده ترکیبی استفاده شده است. در شکل ۶، بخش‌های هدایت کننده مذکور و نحوه ارتباط بین آنها نشان داده شده است. در این معماری، واحد نگاشت تقریبی^۱ شامل اطلاعات کلی موجود درباره ربات و محیط اطراف آن از قبیل مشخصات هندسی زمین مسابقه و ربات، تعداد اشیاء موجود در زمین و غیره می‌باشد. در حقیقت، واحد نگاشت تقریبی با یکپارچه نمودن اطلاعات کلی موجود، مدلی تقریبی از محیط اطراف ربات فراهم می‌آورد. فضای ادراک محلی^۲ اطلاعات محلی جمع‌آوری شده توسط حسگرها و همچنین پیغام‌ها و اطلاعات فرستاده شده از رفتارها را در بانک اطلاعاتی خود نگاه می‌دارد. واحد برنامه‌ریزی^۳ با توجه به اطلاعات موجود در فضای ادراک محلی و واحد نگاشت تقریبی، رفتارهای ربات را برنامه‌ریزی و مدیریت می‌کند. به عبارت دیگر، وظیفه واحد برنامه‌ریزی، فعال و غیرفعال نمودن رفتارهای سیستم بر اساس شرایط محیط و ربات است. بعلاوه، گاهی این واحد، خود مستقیماً کنترل ربات را بر عهده گرفته و آن را هدایت می‌کند. در این حالت، سیستم ترکیبی هدایت ربات به یک سیستم سلسله مراتبی، ساده می‌شود.

وظیفه واحد کنترل، صادر نمودن فرامین لازم به موتورهای محرک ربات و بازو است. واحد حسگرها نیز وظیفه جمع‌آوری اطلاعات حسگرهای مادون قرمز، لامسه و دوربین دیجیتال، و ارسال آنها به فضای ادراک محلی را بر عهده دارد. در حقیقت این دو واحد، پایین‌ترین سطح در سیستم هدایت ربات را تشکیل می‌دهند. دو بخش فوق، عملاً وظیفه ارتباط دیگر بخش‌های نرم‌افزاری با بخش‌های سخت‌افزاری ربات را برعهده دارند.

رفتارها مهمترین بخش هدایت‌کننده ترکیبی مورد استفاده می‌باشند که وظیفه فعال و غیرفعال نمودن آنها برعهده واحد برنامه‌ریزی است. هر رفتار، اطلاعات مورد نیاز خود را از فضای ادراک محلی دریافت می‌کند، در حالی که فرامین صادره برای بخش‌های سخت‌افزاری، مستقیماً به واحد کنترل ارسال می‌شود. در حین اجرای هر رفتار، کلیه فرامین صادره و وضعیت ربات در هر لحظه، در فضای ادراک محلی ثبت می‌گردد. پس از اتمام وظیفه یک رفتار، اجرای آن خاتمه یافته و این موضوع به واحد برنامه‌ریزی اطلاع داده می‌شود. برای ربات طراحی شده، پنج رفتار واکنشی و سه رفتار هدف‌گرا در نظر گرفته شده است. رفتارهای واکنشی عبارتند از:

ره‌گیری حسگری خط هادی (ILF): وظیفه این رفتار، ره‌گیری خط هادی با استفاده از حسگرهای مادون قرمز است. واحد برنامه‌ریزی پس از آنکه ربات را بر روی خط هادی قرار داد، با مشخص نمودن تعداد تقاطع‌هایی که باید طی فرآیند ره‌گیری طی شود (N_c)، این رفتار را فعال می‌کند. ILF با توجه به اطلاعات بدست آمده از حسگرهای فرسوخ، سرعت موتورها را مطابق

¹ Approximation Map

² Local Perceptual Space(LPS)

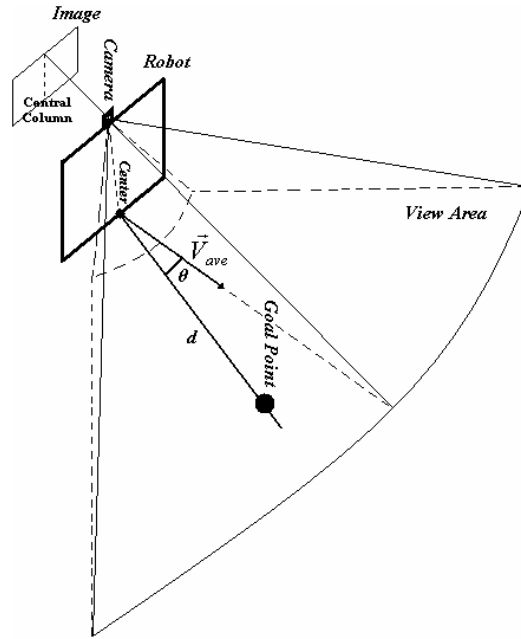
³ Planner

جدول ۱. عملکرد ILF در وضعیت‌های مختلف. منظور از R_c ، R_r و R_l به ترتیب حسگرهای مادون قرمز راست، وسط و چپ است. V_r و V_l نیز به ترتیب بیانگر سرعت موتورهای راست و چپ می‌باشد. همچنین، n_c بیانگر تعداد چهارراه‌های طی شده از زمان فعال شدن ILF است.

R_r	R_c	R_l	V_r (cm/s)	V_l (cm/s)	n_c
OFF	ON	OFF	8	8	No change
OFF	OFF	OFF	8	8	No change
ON	ON	OFF	8	8	No change
OFF	ON	ON	8	8	No change
ON	OFF	OFF	0	8	No change
OFF	OFF	ON	8	0	No change
ON	ON	ON	Robot is located on a cross. If n_c is not smaller than N_c , execution of the behavior is stopped.		Increase n_c by one
ON	OFF	ON			

جدول ۱، به منظور دنبال نمودن خط هادی، تنظیم می‌کند. فرامین لازم برای نگاه داشتن ربات بر روی خط هادی را به موتورهای اعمال می‌کند. با رسیدن ربات به تقاطع N_c -ام، ربات متوقف شده و اجرای ILF نیز خاتمه می‌یابد.

- جستجوی حسگری هدف (IGS): در مقابل هر یک از دروازه‌ها، خط سیاه رنگی عمود بر راستای حرکت ربات قرار گرفته است. واحد برنامه‌ریزی پس از آنکه ربات را در مقابل دروازه‌ها قرار داد، این رفتار را فعال می‌کند. با فعال شدن این فرمان، ربات شروع به حرکت (با سرعت ثابت) رو به جلو (یا عقب) می‌کند. حرکت تا آنجا ادامه می‌یابد که هر سه حسگر مادون قرمز در حالت ON (فعال) قرار گیرند. در این حالت، ربات درست در مقابل دروازه قرار گرفته است لذا، ربات متوقف شده و اجرای IGS خاتمه می‌یابد.
- گرفتن شی (OG): هنگامی که شی در درون بازوی ربات قرار گرفت، واحد برنامه‌ریزی این رفتار را فعال می‌کند. در این حالت، بازوی ربات شروع به بسته شدن می‌کند. این فرآیند تا آنجا ادامه می‌یابد که حسگرهای لامسه بازو فعال گردند.
- رهاکردن شی (OD): هنگامی که بازوی ربات درون یکی از دروازه‌ها قرار می‌گیرد، واحد برنامه‌ریزی این رفتار را فعال می‌کند. در این حالت، بازوی ربات شروع به باز شدن می‌کند. بازو تا آنجا باز می‌شود که حسگرهای لامسه آن غیرفعال گردند.
- رفتار جستجوی تصویری هدف (VGS)، تنها رفتار واکنشی است که از اطلاعات تصویری استفاده می‌کند. با فعال شدن این رفتار، ربات در تصاویر دریافتی از دوربین به جستجوی اشیاء موجود در زمین می‌پردازد. برای این منظور، از روش فیلترهای منطبق رنگ که در بخش ۵ به تفصیل شرح داده خواهد شد، استفاده گردیده است. به محض یافتن اولین شی، مختصات (X_0) و نوع (t_0) آن در فضای ادارک محلی ذخیره و اجرای این رفتار خاتمه می‌یابد.
- رفتارهای هدف‌گرای ربات همگی برپایه استفاده از اطلاعات تصویری بنا نهاده شده‌اند. رفتار تعقیب تصویری هدف (VGF)، مهمترین رفتار هدف‌گرای ربات است. هسته مرکزی VGF را یک کنترل کننده PD-P با پاسخ‌دهی تصویری تشکیل می‌دهد. کنترل کننده مذکور در هر لحظه، با محاسبه خطای زاویه و فاصله ربات با استفاده از الگوریتم‌های پردازش تصویر، به تصحیح سرعت چرخ‌ها می‌پردازد (در بخش ۵، کنترل کننده دیداری به تفصیل شرح داده خواهد شد). VGF زیربنای دو رفتار هدف‌گرای دیگر ربات را نیز تشکیل می‌دهد:
- دنبال نمودن تصویری خط هادی (VLF): واحد برنامه‌ریزی، این رفتار را زمانی فعال می‌کند که خط هادی در محدوده دید ربات قرار گرفته باشد. در این حالت، با پردازش تصاویر دریافتی (با استفاده از فیلترهای منطبق رنگ)، خط هادی در تصویر شناسایی شده، مرکز ثقل آن (X_L) به عنوان نقطه هدف به VGF معرفی می‌شود. با خارج شدن خط هادی از محدوده دید ربات (که ممکن است ناشی از رسیدن ربات به انتهای خط هادی باشد)، اجرای این رفتار نیز خاتمه می‌یابد.
- شکار تصویری هدف (VGH): پس از آنکه VGS شی را در تصویر شناسایی نمود، واحد برنامه‌ریزی این رفتار را به منظور شکار آن فعال می‌کند. این رفتار با استفاده از روش فیلترهای منطبق رنگ، شی هدف را در تصویر شناسایی نموده و مختصات مرکز ثقل آن را به عنوان نقطه هدف در اختیار VGF قرار می‌دهد. در هر مرحله از اجرای این رفتار، مختصات مرکز ثقل شی، در



شکل ۸. نمایی از ربات و شی هدف

۵) کنترل کننده دیداری

وظیفه کنترل کننده دیداری، کنترل حرکت ربات به سوی هدف مورد نظر است. این کنترل کننده بی‌درنگ^۱، پس از دریافت تصویری از صحنه پیش روی ربات و انجام پردازش‌های مناسب (که در بخش ۶ به تفصیل شرح داده خواهد شد)، نقطه منتخب شی هدف در تصویر را استخراج کرده و سپس حرکت ربات را در جهت رسیدن به آن، کنترل می‌کند. فرض کنید، مطابق شکل ۸، زاویه بین راستای حرکت و خط واصل ربات به هدف و d فاصله آن تا هدف باشد. در این حالت، معادلات حاکم بر حرکت ربات به شرح ذیل است [۸]:

$$\begin{cases} V_{ave}(t) = \frac{1}{2}(V_r(t) + V_l(t)) \\ \dot{d}(t) = V_{ave} \cos(\theta) \\ \dot{\theta}(t) = \frac{1}{L}(V_r - V_l) \end{cases} \quad (1)$$

که در آن، L فاصله بین دو چرخ، $V_r(t)$ سرعت چرخ راست، $V_l(t)$ سرعت چرخ چپ و $V_{ave}(t)$ سرعت مرکز ربات در لحظه t است. با نزدیک شدن ربات به هدف، θ و d به صورت توأم به سمت صفر میل می‌کنند. بنابراین، در هر لحظه می‌توان برای ربات دو نوع خطا در نظر گرفت: خطای زاویه θ و خطای فاصله d .

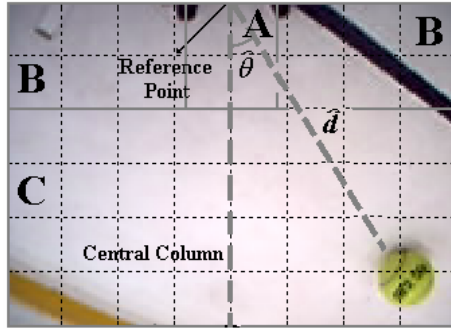
۵-۱) اندازه‌گیری خطا

ما خطای زاویه و فاصله را با استفاده از الگوریتم‌های پردازش تصویر تخمین زده‌ایم. همانطور که در شکل ۹ نشان داده شده، با دقت بالایی می‌توان گفت که راستای حرکت ربات منطبق بر ستون مرکزی^۲ تصویر است. بر این اساس، در هر تصویر، پیکسل وسط از ردیف اول که نقطه مرجع^۳ نامیده می‌شود، نزدیکترین پیکسل به مرکز ربات می‌باشد. در شکل ۹، زاویه بین ستون مرکزی و خط واصل نقطه مرجع به هدف در تصویر با $\hat{\theta}$ و فاصله هدف تا نقطه مرجع با \hat{d} نشان داده شده است. به آسانی می‌توان نشان داد که $\hat{\theta}$ و \hat{d} به ترتیب با θ و d دارای رابطه‌ای غیر خطی هستند:

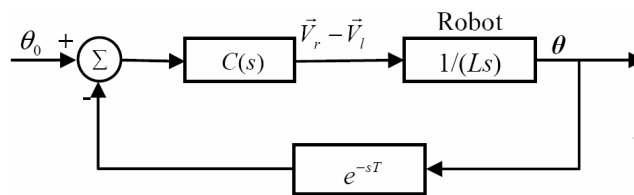
¹ On-line

² Central column

³ Reference point



شکل ۹. تصویری از صحنه پیش روی ربات. در این تصویر، ستون مرکزی، نقطه مرجع، زاویه بین ستون مرکزی و خط واصل نقطه مرجع به هدف ($\hat{\theta}$ به عنوان تخمینی از خطای زاویه)، و فاصله نقطه مرجع تا هدف (\hat{d} به عنوان تخمینی از خطای فاصله)، نشان داده شده است.



شکل ۱۰. نمودار بلوکی سیستم کنترل اختلاف سرعت بین دو چرخ ربات به منظور تصحیح زاویه حرکت آن.

$$\begin{cases} \hat{\theta} = f(\theta, d) \\ \hat{d} = g(\theta, d) \end{cases} \quad (2)$$

تعیین دقیق توابع f و g نیازمند استفاده از روش‌های واسنجی دوربین جهت تخمین مقادیر پارامترهای داخلی و خارجی دوربین می‌باشد [۱۷]. اما با در نظر گرفتن نکات ذیل می‌توان از فرآیند واسنجی صرف نظر نموده، از $\hat{\theta}$ و \hat{d} به عنوان تخمینی از خطای زاویه و فاصله استفاده نمود:

- توابع f و g به ترتیب نسبت به تغییرات θ و d ، توابعی صعودی هستند. بدین معنی که با افزایش/کاهش θ (با فرض ثابت بودن d و \hat{d}) یا فرض ثابت بودن θ ، به ترتیب $\hat{\theta}$ و \hat{d} نیز افزایش/کاهش می‌یابند.
- چنانچه θ و d صفر شوند، $\hat{\theta}$ و \hat{d} نیز صفر خواهند شد.

مشاهدات تجربی نشان می‌دهد که کنترل کننده طراحی شده برای کنترل حرکت ربات، در برابر اعوجاجات ناشی از تخمین‌های فوق مقاوم می‌باشد.

۲-۵) تصحیح خطای زاویه

همانطور که در دستگاه معادلات ۱ نشان داده شده است، اختلاف سرعت بین دو چرخ ربات، متناسب با سرعت زاویه‌ای ربات ($\dot{\theta}$) است. بنابراین برای تصحیح θ ، باید کنترل کننده‌ای طراحی نمود که در هر لحظه $V_r - V_l$ را به شکل مناسبی تنظیم نماید. نمودار بلوکی سیستم کنترلی در شکل ۱۰ نشان داده شده است. در این نمودار، e^{-sT} بیانگر تأخیر ناشی از زمان محاسباتی لازم برای پردازش یک تصویر و $C(s)$ تابع تبدیل کنترل کننده است. با توجه به اینکه در این حالت، ربات به صورت یک سیستم انگرال‌گیر عمل می‌کند، از یک کنترل کننده تناسبی-مشتق‌گیر (PD) برای تصحیح خطای زاویه ربات استفاده شده است. لذا داریم:

$$C(s) = k_p + k_d s \quad (3)$$

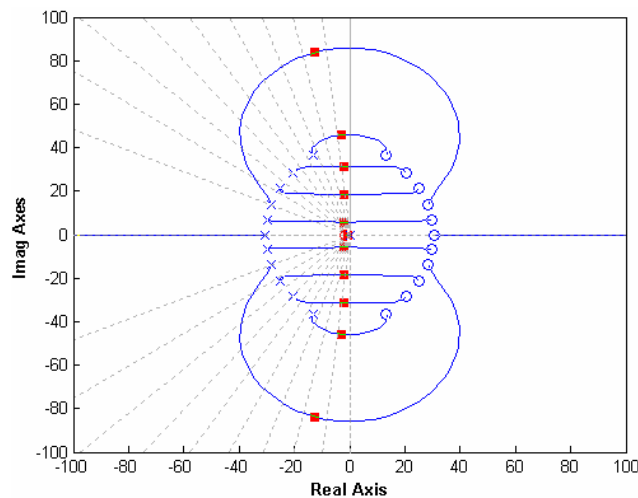
برای تنظیم ضرایب کنترل کننده فوق، از محیط Simulink (در نرم‌افزار Matlab) استفاده گردیده است. با توجه به اینکه زمان محاسباتی هر تصویر $T = 0.3 \text{ sec}$ و فاصله بین دو چرخ $L = 14 \text{ cm}$ می‌باشد، مطابق شبیه‌سازی‌های انجام شده و مشاهدات تجربی، مقادیر $4/6$ و $2/6$ به ترتیب برای ضرایب k_p و k_d بدست آمد. به منظور بررسی پایداری سیستم با ضرایب فوق، نمودار مکان هندسی

ریشه‌های سیستم حلقه بسته مطابق شکل ۱۱ بررسی شده است. برای این منظور، تأخیر غیرخطی موجود در شاخه پسخورد (شکل ۱۰) با استفاده از تقریب Padé [۱۸] به صورت خطی تخمین زده شده است. مطابق شکل، سیستم حلقه بسته با استفاده از کنترل کننده PD (کنترل کننده زاویه) پایدار بوده و صفر شدن زاویه خطای θ را تضمین می‌کند.

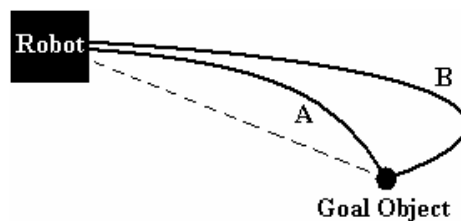
۵-۳) تصحیح خطای فاصله

مطابق دستگاه معادلات ۱، مشتق خطای فاصله (d) با سرعت حرکت مرکز ربات (V_{ave}) رابطه مستقیم دارد. بنابراین برای صفر نمودن خطای d کافی V_{ave} تصحیح شود. با تعیین مقدار V_{ave} و در دست داشتن $V_r - V_l$ (با استفاده از کنترل کننده زاویه) می‌توان سرعت هر چرخ را به طور جداگانه در هر لحظه بدست آورد. به نظر می‌رسد، کنترل کننده تناسبی یک انتخاب مناسب برای این منظور باشد.

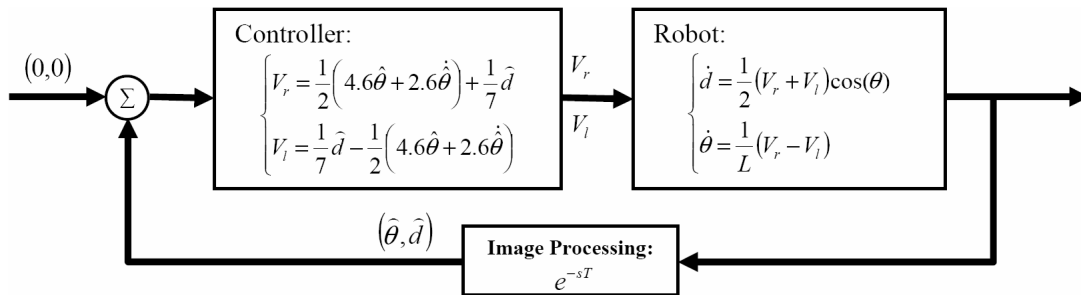
همانطور که قبلاً شرح داده شد، کنترل کننده زاویه تنها اختلاف سرعت بین دو چرخ ربات را تعیین می‌کند. با اینکه سیستم با استفاده از این کنترل کننده تحت هر شرایطی پایدار است؛ اما عملکرد آن، به مقدار V_{ave} بستگی دارد. مطابق شکل ۱۲، چنانچه V_{ave} بیش از حد بزرگ باشد، فراجاهش در سیستم رخ می‌دهد. در این حالت، کنترل کننده قادر نیست قبل از رسیدن ربات به هدف، خطای زاویه را تصحیح نماید. بنابراین ربات ابتدا از هدف عبور کرده، و سپس به سمت آن دور می‌زند (نمودار B در شکل ۱۲). با توجه به اینکه دامنه دید دوربین محدود است، فراجاهش سبب می‌شود که شی هدف از دامنه دید دوربین خارج گردد. در این حالت، دیگر سیستم قادر به تخمین خطا نخواهد بود. لذا، در تعیین مقدار V_{ave} ، علاوه بر فاصله ربات تا هدف، خطای زاویه نیز باید مد نظر قرار گیرد.



شکل ۱۱. نمودار مکان هندسی ریشه‌های سیستم شکل ۱۰. در این شکل قطب‌های سیستم با مربع‌های توپر قرمز رنگ نشان داده شده است.



شکل ۱۲. عملکرد کنترل کننده تصحیح خطای زاویه در دو وضعیت مختلف. در وضعیت A، سرعت مرکز ربات (V_{ave}) مناسب بوده، در حالی که در وضعیت B سرعت مرکز ربات بیش از اندازه زیاد بوده که در نتیجه آن، سیستم دچار فراجاهش گشته است.



شکل ۱۳. نمودار بلوکی سیستم نهایی کنترل ربات.

برای غلبه بر مشکل فوق، مطابق شکل ۹، تصاویر دریافتی از دوربین به سه ناحیه مجزا A ، B و C تقسیم می‌گردند. چنانچه شی هدف در ناحیه C قرار گیرد، V_{ave} با استفاده از کنترل کننده تناسبی $V_{ave} = 1/7 \times d$ تعیین می‌شود. اگر شی در ناحیه B قرار گرفته باشد، $V_{ave} = 0^{cm/s}$ خواهد بود. در این حالت، ربات به صورت درجا شروع به چرخش می‌کند (برای تصحیح خطای زاویه)، تا اینکه شی در یکی از نواحی B یا A قرار گیرد. در صورتی که شی در ناحیه A قرار گیرد، $V_{ave} = 6^{cm/s}$ و $V_r - V_l = 0^{cm/s}$ خواهد بود. به عبارت دیگر، در این حالت کنترل کننده‌های زاویه و فاصله غیر فعال شده و ربات با سرعت ثابت (طی مدت زمان مشخص) به سمت جلو حرکت می‌کند تا شی در درون بازوی ربات قرار گیرد. نمودار بلوکی سیستم نهایی کنترل ربات در شکل ۱۳ نشان داده شده است.

۶) پردازش تصویر

تشخیص همزمان اشیا در تصویر، مستلزم یک الگوریتم قسمت‌بندی مناسب با قابلیت اطمینان و انعطاف بالا است. در مساله مورد نظر ما مهمترین و بهترین مشخصه اشیا، رنگ و شکل متفاوت آنهاست. استفاده از روش‌های آشکارسازی مبتنی بر شکل، بسیار زمانبر هستند. خصوصاً اشیاایی چون جعبه صابون و قوطی نوشابه، در حالات مختلف (مثلاً افتاده به پهلو یا ایستاده) اشکال متفاوتی دارند، که این خود دشواری کار را دوچندان می‌کند. بنابراین ویژگی رنگ، مبنای شناسایی اشیا در این مقاله قرار گرفته است. هرچند که در استفاده از ویژگی رنگ نیز تمهیداتی چون عدم وابستگی الگوریتم به روشنایی محیط باید در نظر گرفته شود.

تاکنون الگوریتم‌های مختلفی برای تشخیص رنگ و ناحیه‌بندی تصاویر رنگی ارائه شده است. به عنوان مثال، Chen و همکارانش [۱۹] از اطلاعات موجود در کنتراست تصویر به جای مشتقات آن، برای ناحیه‌بندی تصاویر رنگی استفاده کرده‌اند. در الگوریتم پیشنهادی Shafarenko و همکارانش [۲۰]، پس از حذف نویز تصویر، هیستوگرام سه بعدی رنگ در فضای رنگی LUV محاسبه می‌شود. سپس، از یک روش خوشه‌بندی بر اساس تبدیل حوضچه ریخت‌شناسی^۱ برای ناحیه‌بندی تصویر استفاده گردیده است. همچنین، همان نویسندگان در کار دیگری، با تعریف یک معیار گرادیان رنگ جدید در همان فضای رنگی برای تبدیل حوضچه، روش جدیدی برای ناحیه‌بندی تصاویر رنگی ارائه داده‌اند [۲۱]. به عنوان مثالی دیگر، Gevers [۲۲] از گرادیان‌های رنگی نسبت-ثابت^۲ برای ناحیه‌بندی تصاویر بافتی استفاده کرده است. اخیراً، Yang و همکارانش [۲۳]، برای تعریف نگاشت لبه در روش مرز فعال جریان بردار گرادیان^۳ از گرادیان رنگ در فضای رنگی LUV استفاده کرده‌اند. آنها روش فوق را برای ناحیه‌بندی سلول‌های خونی بکار گرفته‌اند.

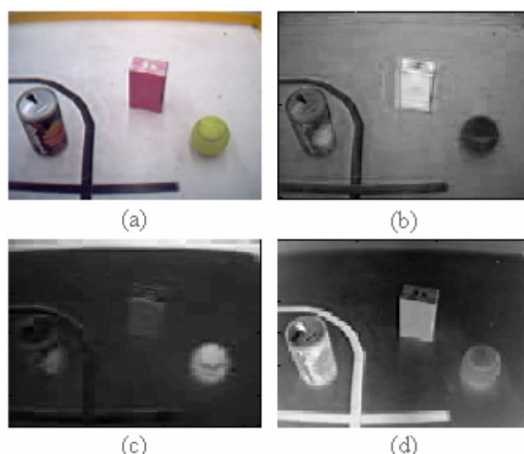
همه الگوریتم‌های تشخیص رنگ، به مسأله تغییر مؤلفه‌های رنگی در اثر تغییر روشنایی محیط، توجه ویژه‌ای داشته‌اند [۲۴]. در حقیقت، رنگ در تصویر تنها بیانگر خصوصیات طیفی نور نبوده و به شدت روشنایی آن نیز وابسته است. در نتیجه استخراج ویژگی‌های رنگی مقاوم در برابر تغییرات روشنایی محیط، امری پیچیده و دشوار است. در این مقاله، برای آشکارسازی رنگ، روش جدیدی تحت عنوان فیلتر منطبق رنگ^۴ (CMF) معرفی شده است.

¹ Morphological watershed transform

² Color constant ratio gradients

³ Gradient vector flow

⁴ Color Match Filter (CMF)



شکل ۱۴. (a) تصویر رنگی پیش از پردازش و تصاویر حاصل از اعمال فیلترهای (b) قرمز، (c) زرد و (d) سیاه.

۱-۶ فیلتر منطبق رنگ

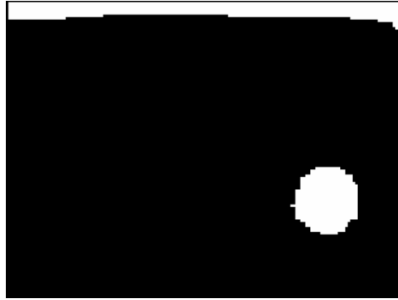
نور مرئی صورتی از انرژی الکترومغناطیسی است. نور دو اثر مهم بر چشم دارد: روشنایی که نشان‌دهنده تراکم طول موج‌های مختلف نور بوده و رنگ که بیانگر ترکیب طیفی آن می‌باشد. نظریه یانگ توضیح می‌دهد که نور را می‌توان بجای تابع کاملی از طول موج‌های مختلف در محدوده مرئی، تنها با سه عدد مشخص کرد. نشان داده شده است که بردارهای سه‌تایی رنگ تشکیل یک فضای برداری می‌دهند [۲۵]. تاکنون دستگاه‌های مختصات متعددی برای بیان رنگ ارائه شده است که از آنجمله می‌توان به RGB, HSV, XYZ, LUV و LAB اشاره نمود. در تمام فضاهای رنگی فوق بجز RGB، رنگ با سه مؤلفه که به ترتیب بیانگر نوع رنگ، شدت رنگ و روشنایی هستند، بیان می‌شود. امروزه، از دستگاه‌های مختصات رنگ در کاربردهای گوناگون پردازش رنگ استفاده می‌شود [۲۶-۲۸]. در این مقاله از دستگاه مختصات RGB به جهت نزدیکی آن به دستگاه‌های تصویربرداری استفاده شده است. براین اساس، مطابق معادله ذیل برای بردار رنگ $X = (x_r, x_g, x_b)$ ، بردار یکه $F = (f_r, f_g, f_b)$ چنان تعریف می‌شود که همبستگی میان این دو بردار بیشینه گردد:

$$K(F, X) = F \cdot X = \frac{1}{|X|} (f_r x_r + f_g x_g + f_b x_b) \quad (۴)$$

که در آن، عملگر $|\cdot|$ اندازه یک بردار یا تعداد عناصر یک مجموعه را بر می‌گرداند. بردار یکه F را فیلتر منطبق رنگ می‌نامیم که f_r ، f_g و f_b ضرایب فیلتر مذکور به ترتیب در باندهای قرمز، سبز و آبی هستند. بدیهی است که $|F| = 1$. به عبارت دیگر ایده اصلی فیلترهای منطبق رنگ بر این اساس استوار است که می‌توان با ترکیب مناسب باندهای سبز، قرمز و آبی در تصویر، رنگ خاصی را نسبت به دیگر رنگ‌های موجود در صحنه تصویربرداری تقویت نمود. به عنوان مثال، در شکل ۱۴-a صحنه‌ای شامل یک توپ و دیواره زرد رنگ، یک جعبه صابون قرمز رنگ، و یک خط و قوطی نوشابه سیاه رنگ نشان داده شده است. همانطور که در شکل‌های ۱۴-b، ۱۴-c و ۱۴-d می‌توان مشاهده نمود، با اعمال فیلترهای $(r=0.5, g=0.5, b=-1)$ ، $(r=1, g=-1, b=0.3)$ ، $(r=-1, g=0, b=0)$ بر تصویر ۱۲-a، به ترتیب رنگ‌های زرد، قرمز و سیاه تقویت و آشکار شده‌اند. برای حذف نسبی اثر نور محیط، می‌توان از دستگاه مختصات RGB-نرمالیزه شده، مطابق معادله ذیل استفاده نمود:

$$\left(R_n = \frac{R}{R+G+B}, G_n = \frac{G}{R+G+B}, B_n = \frac{B}{R+G+B} \right) \quad (۵)$$

مشاهدات تجربی نشان داد که حتی حذف مؤلفه روشنایی نیز نمی‌تواند اثر تغییرات نور محیط را به طور کامل حذف نماید. بعلاوه شناسایی رنگ‌هایی چون خاکستری و سیاه تنها با استفاده از دو مؤلفه رنگی دستگاه مختصات رنگ، امکان‌پذیر نیست.



شکل ۱۵. نواحی زرد رنگ در تصویر a-۱۴.

روش پیشنهادی در این مقاله برای غلبه بر مشکلات فوق بدین شرح است. ابتدا از صحنه‌های مورد نظر، تحت شرایط نوری مختلف تصویربرداری می‌شود. سپس از رنگ‌های موجود در این تصاویر نمونه‌برداری می‌گردد. از کنار هم قرار گرفتن تمام نمونه‌های هم‌رنگ، تصویر مرجعی بوجود می‌آید که شامل تمام رنگ‌های موجود در صحنه، تحت شرایط نوری مختلف می‌باشد. بدیهی است که بخشی از تصویر مرجع شامل نمونه‌های استخراج شده از رنگ مورد نظر ما، تحت شرایط نوری مختلف است. معادله فیلتر منطبق نیز طبق معادله (۶) توسعه می‌یابد. در حقیقت، در معادله ذیل، فیلتر منطبق به صورت ترکیبی از مؤلفه‌های رنگی و روشنایی تعریف شده است.

$$\begin{cases} \mathbf{F} = (f_r, f_g, f_b, f_{nr}, f_{ng}, f_{nb}) \\ K(\mathbf{F}, \mathbf{X}) = \frac{1}{|\mathbf{X}|} \left(f_r x_r + f_g x_g + f_b x_b + \frac{f_{nr} x_r + f_{ng} x_g + f_{nb} x_b}{x_r + x_g + x_b} \right) \end{cases} \quad (6)$$

اکنون این سؤال مطرح است که بهترین فیلتر برای تقویت یک رنگ خاص در یک صحنه چیست؟ و چگونه می‌توان این فیلتر را بدست آورد؟ ما برای پاسخ به این سؤال از الگوریتم‌های ژنتیکی استفاده کرده‌ایم. الگوریتم پیشنهادی در دو مرحله اجرا می‌گردد. ابتدا در مرحله طراحی، فیلتر بهینه برای آشکارسازی رنگ مورد نظر، با استفاده از الگوریتم‌های ژنتیکی استخراج می‌شود. سپس در مرحله بازرسی، با اعمال فیلتر بهینه مذکور بر تصویر، نواحی که دارای رنگ مورد نظر هستند، آشکار می‌گردند.

۶-۲) طراحی تکاملی فیلترهای منطبق رنگ

الگوریتم‌های ژنتیکی خانواده‌ای از مدل‌های محاسباتی هستند که از تکامل موجودات (مسأله انتخاب طبیعی در طبیعت) الهام گرفته شده [۲۹] و اولین بار توسط Holland پیشنهاد گردیده‌اند [۳۰]. انتخاب طبیعی یک فرایند زیستی است که بر اساس آن در یک جامعه رقابتی، احتمال برنده شدن افراد قویتر بیشتر است. امروزه از الگوریتم‌های ژنتیکی در کاربردهای مختلفی مانند بهینه‌سازی توابع، شناسایی سیستم‌ها، پردازش تصویر و... استفاده می‌شود [۲۹]. هنگامی که هدف، پیدا کردن جوابی نزدیک به پاسخ بهینه، برای یک تابع چندبعدی با چندین قید باشد، بهینه‌سازهای ژنتیکی بسیار مؤثر و کارا هستند [۳۱].

اولین مسأله در معماری یک الگوریتم ژنتیکی، کدگذاری^۱ مناسب مسأله است. در الگوریتم ژنتیک پیشنهادی، هر کروموزوم بیانگر یک فیلتر منطبق رنگ است. بنابراین، هر کروموزوم از شش ژن تشکیل شده است (به معادله ۶ مراجعه شود). ژن‌ها (یا همان ضرایب فیلتر) نیز اعدادی حقیقی در بازه [-1,1] می‌باشند. دومین گام تعریف تابع هزینه، برای سنجش کیفیت فیلترها است. بدین منظور، نخست مناطقی از تصویر مرجع که دارای رنگ مورد نظر هستند، مشخص می‌شوند. به عنوان مثال در شکل ۱۵، نواحی زرد رنگ تصویر a-۱۴ نشان داده شده است. هدف فرآیند تکاملی، بیشینه نمودن اختلاف پاسخ فیلتر در این مناطق نسبت به سایر بخش‌های تصویر است.

فرض کنید تصویر مرجع C به بلاک‌های $L \times L$ تقسیم شده است. به طوری که B_i ($i = 1, 2, \dots, N$) بیانگر بلاک i -ام و N تعداد کل بلاک‌ها باشد. همانطور که در معادله ذیل نشان داده شده است، بلاک‌های B_i با یکدیگر هیچ نقطه مشترکی نداشته و تمام سطح تصویر مرجع C را می‌پوشانند.

^۱ Coding

$$\bigcup_{i=1}^N B_i = C \quad (7)$$

$$B_i \cap_{i \neq j} B_j = \phi, \quad i, j = 1, 2, \dots, N \quad (8)$$

همچنین طبق معادله‌های (۹) و (۱۰)، C_b و C_f به ترتیب مجموعه پیکسل‌های دارای رنگ X و فاقد آن در تصویر مرجع C هستند.

$$C_f = \{(x, y) | C(x, y) = X\} \quad (9)$$

$$C_b = \{(x, y) | C(x, y) \neq X\} \quad (10)$$

با استفاده از بلاک‌های B_i مجموعه C_b به مجموعه‌های $C'_b(i)$ افزای می‌شود؛ به طوری که:

$$C'_b(i) = B_i \cap C_b, \quad i = 1, 2, \dots, N \quad (11)$$

$$\bigcup_{i=1}^N C'_b(i) = C \quad (12)$$

$$C'_b(i) \cap_{i \neq j} C'_b(j) = \phi, \quad i, j = 1, 2, \dots, N \quad (13)$$

بر این اساس، تابع هزینه الگوریتم ژنتیک پیشنهادی برای فیلتر F به شکل زیر تعریف می‌شود:

$$E(F) = \frac{1}{|C_f|} \sum_{(x,y) \in C_f} K(F, C(x, y)) - \max_{i=1}^N \left[\frac{1}{|C'_b(i)|} \sum_{(x,y) \in C'_b(i)} K(F, C(x, y)) \right] \quad (14)$$

در تابع هزینه فوق، اختلاف میانگین پاسخ فیلتر در مناطق با رنگ X از تصویر مرجع C ، نسبت به سایر مناطق محاسبه می‌شود. بدیهی است که هرچه اختلاف فوق بیشتر گردد، فیلتر منطبق‌تر است، به شکل مناسب‌تری رنگ X را از سایر رنگ‌ها تفکیک کند. تابع هزینه پیشنهادی دارای سه ویژگی مهم است:

- هر فیلتر متناسب با پاسخش به رنگ X در مقایسه با سایر رنگ‌های صحنه، تشویق می‌شود.
- هر فیلتر متناسب با پاسخش به سایر رنگ‌های صحنه (غیر از رنگ X) و تعداد پیکسل‌های آنها جریمه می‌شود.
- با تقسیم تصویر به بلاک‌های B_i ، عملاً پاسخ محلی فیلتر، در ارزشیابی آن مد نظر قرار می‌گیرد.

در معادله (۱۴)، چنانچه L (اندازه بلاک‌های B_i) خیلی بزرگ باشد، از دقت الگوریتم ژنتیکی در طراحی کاسته می‌شود. در حالی که برعکس، اگر L خیلی کوچک انتخاب شود، پاسخ‌های الگوریتم به شدت تحت تأثیر نویز قرار می‌گیرد. از اینرو در انتخاب L باید مصالحه‌ای بین دقت الگوریتم ژنتیکی و حذف اثر نویز انجام پذیرد. یک روش مناسب برای انتخاب ابعاد بلاک‌ها، انتخاب بر اساس مساحت کوچکترین شی موجود در تصویر می‌باشد. ما در شبیه‌سازی‌ها، مقدار ۲۵ را برای پارامتر L در نظر گرفته‌ایم که تقریباً برابر با کوچکترین شی موجود در تصویر است.

تکامل بیش از آنکه به حفظ گونه‌هایی بپردازد که تطابق بیشتری با محیط اطرافشان داشته‌اند، به حذف گونه‌هایی پرداخته که کمتر با محیط اطرافشان تطبیق داشته‌اند [۳۲]. به عبارت دیگر حذف عناصر ضعیفتر به مراتب اثر بیشتری از حفظ عناصر قویتر در تکامل دارد. در این الگوریتم ژنتیکی در هر تولید نسل^۱، از میان جمعیت موجود تنها دو کروموزوم به عنوان ولی انتخاب می‌شوند و بنابراین تنها دو کروموزوم فرزند تولید می‌گردد. جمعیت جدید با جایگذاری این دو فرزند (به شرط آنکه در جمعیت قبلی کروموزومی مشابه آنها نباشد) بجای بدترین کروموزوم‌های جمعیت قبلی (آنهايي که بدترین درجه شایستگی را دارند) بوجود می‌آید. همانطور که مشاهده می‌شود، ما در حقیقت از مدل Genitor [۳۳] استفاده کرده‌ایم. همانطور که گفته شد، در هر تولید نسل، باید دو ولی از میان جمعیت فعلی انتخاب گردد. یکی از آنها را بروش مسابقه^۲ [۳۴] و دیگری را به صورت تصادفی انتخاب می‌کنیم. هدف از این نحوه انتخاب والدین، افزایش سرعت همگرایی الگوریتم ژنتیکی و حفظ پراکندگی جمعیت در فضای جستجو است. با تعیین عملگرهای برش و جهش، الگوریتم ژنتیکی مورد نظرمان کامل می‌گردد. با این فرض که بردارهای P_1 و P_2 ، دو کروموزوم ولی و a عددی تصادفی باشد، عملگر برش را طبق معادله ذیل تعریف می‌کنیم:

¹ Generation

² Tournament

$$C_1 = a \times P_1 + (1-a) \times P_2 \quad (15)$$

$$C_2 = (1-a) \times P_1 + a \times P_2 \quad (16)$$

که در آن، C_1 و C_2 به ترتیب کروموزوم‌های فرزند هستند. از عملکرد جهش برای فرار از ماکزیمم‌های محلی استفاده می‌شود. قبل از اینکه الگوریتم ژنتیکی شروع بکار کند، عددی به عنوان احتمال جهش در نظر گرفته می‌شود. در طول فرآیند تکامل، برای هر کروموزوم فرزند، عددی تصادفی تولید می‌شود. اگر این عدد از احتمال جهش کوچکتر بود، فرزند مذکور کاملاً به صورت تصادفی از فضای جستجو، انتخاب و در غیر این صورت بدون تغییر باقی می‌ماند. ثابت گردیده است که چنانچه در یک الگوریتم ژنتیکی، بهترین کروموزوم‌های نسل قبلی با احتمال بالا به نسل جدید وارد شوند، الگوریتم به سمت بهینه مطلق همگرا می‌شود [۳۵]. در الگوریتم ژنتیکی پیشنهادی که آن را GACFID^۱ می‌نامیم، همواره بهترین کروموزوم نسل فعلی در نسل بعدی نیز قرار می‌گیرد. بنابراین، می‌توان گفت که چنانچه این الگوریتم به تعداد کافی تکرار شود، می‌تواند بهترین فیلتر منطبق را بیابد. نتایج تجربی بیانگر آن است که الگوریتم ژنتیکی پیشنهادی، در عمل نیز عملکرد بسیار مناسبی از خود نشان می‌دهد.

۳-۶ ناحیه‌بندی تصویر با استفاده از فیلترهای منطبق رنگ

در بخش قبل، فیلترهای منطبق رنگ با استفاده از یک الگوریتم تکاملی طراحی گردیدند. مطابق معادله (۱۷)، با اعمال فیلتر منطبق F (که برای آشکارسازی رنگ X طراحی شده است) بر تصویر رنگی I ، یک تصویر سطح خاکستری بدست می‌آید.

$$I_f(x, y) = K(F, I(x, y)) \quad (17)$$

با توجه به روش طراحی فیلتر F انتظار می‌رود که در تصویر I_f ، نواحی با رنگ X نسبت به سایر نواحی دارای سطوح خاکستری بزرگتری باشند. اکنون باید با انتخاب یک مقدار آستانه مناسب نواحی مذکور، از تصویر استخراج گردند. قبل از انتخاب مقدار آستانه، مطابق معادله‌های (۱۸) و (۱۹) به ترتیب نویز تصویر I_f با اعمال فیلتر میانه تضعیف گردیده (\bar{I}_f) و سطوح خاکستری تصویر حاصل نیز در بازه $[0, 1]$ نرمالیزه می‌شود (\bar{I}_f).

$$\bar{I}_f = \text{median}(I_f) \quad (18)$$

$$\bar{I}_f(x, y) = \frac{\bar{I}_f(x, y) - \min(\bar{I}_f)}{\max(\bar{I}_f) - \min(\bar{I}_f)} \quad (19)$$

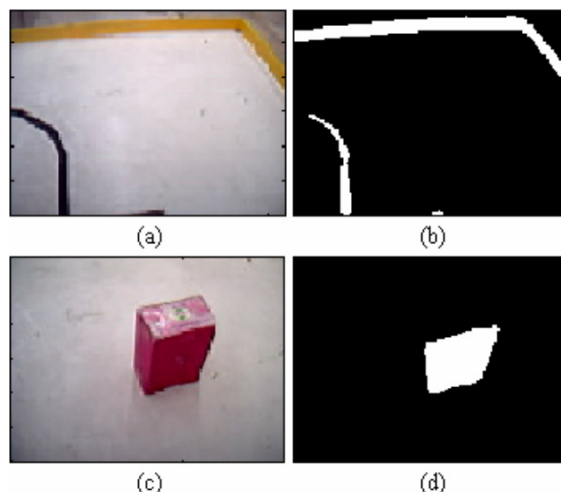
از آنجا که ممکن است پاسخ فیلتر منطبق، وابسته به شرایط نوری صحنه تصویربرداری باشد، لذا مقدار آستانه در تصویر \bar{I}_f ، با استفاده از روش انتخاب آستانه به صورت تکراری^۲ (ITS) [۳۶] محاسبه می‌شود:

$$\eta = \text{ITS}(\bar{G}) \quad (20)$$

در تصویر \bar{I}_f ، تمام پیکسل‌هایی که سطح خاکستری آنها بیشتر از مقدار آستانه η باشد، دارای رنگ X هستند. انتخاب آستانه با این روش، انعطاف‌پذیری زیادی به الگوریتم قسمت‌بندی تصویر می‌بخشد. نتایج تجربی نشان می‌دهند، روش فوق در مواردی که شی با رنگ X در تصویر وجود دارد، عملکرد بسیار مناسبی از خود نشان می‌دهد. اما در سایر موارد، الگوریتم به اشتباه، نواحی دیگری را آشکار می‌کند. به عنوان مثال، در شکل ۱۶-ا، شی با رنگ قرمز وجود ندارد. با اعمال فیلتر منطبق رنگ قرمز بر این تصویر (شکل ۱۶-ب)، به اشتباه، خط مشکی و دیواره زرد رنگ آشکار شده‌اند. در حالی که با اعمال همان فیلتر منطبق بر شکل ۱۶-ج، جعبه صابون قرمز رنگ بخوبی شناسایی شده است (شکل ۱۶-د). دلیل بروز این مشکل در شیوه انتخاب آستانه است. روش ITS، آستانه را چنان تنظیم می‌کند که از میان رنگ‌های موجود در تصویر، رنگی که نسبت به بقیه بیشتر تقویت گردیده، استخراج شود. بنابراین چنانچه در تصویر، شی با رنگ X وجود نداشته باشد، الگوریتم قادر به تشخیص این موضوع نبوده و در نتیجه، فرآیند ناحیه‌بندی دچار اشتباه خواهد شد. این مشکل را می‌توان با ایجاد یک شی مجازی با رنگ X در تصویر، حل نمود.

^۱ GA Color Filter Designer (GACFID)

^۲ Iterative Threshold Selection (ITS)



شکل ۱۶. نمونه‌ای از عملکرد الگوریتم ناحیه‌بندی با استفاده از فیلترهای منطبق رنگ. چنانچه در تصویر شی قرمز رنگی وجود نداشته باشد، پاسخ الگوریتم ناحیه‌بندی اشتباه خواهد بود. این مشکل با اضافه کردن یک شی مجازی قرمز رنگ به تصویر رفع می‌شود. (a) تصویری از یک صحنه بدون رنگ قرمز و (b) پاسخ الگوریتم (بدون در نظر گرفتن شی مجازی قرمز رنگ در تصویر) پس از اعمال فیلتر منطبق رنگ قرمز. (c) تصویری از یک صحنه شامل یک شی قرمز رنگ و (d) نواحی قرمز رنگ آشکار شده پس از اعمال فیلتر منطبق رنگ.

همانطور که مشاهده می‌شود، برخلاف مرحله طراحی، مرحله ناحیه‌بندی از حجم محاسباتی کمی برخوردار است. باید توجه داشت که از دیدگاه ربات، فرآیند طراحی فیلترها یک فرآیند خارج خط^۱ است و لذا محدودیت زمانی چندانی ندارد. اما آنچه اهمیت دارد این است که پس از طراحی فیلترها، در حین بازرسی تصاویر، الگوریتم ناحیه‌بندی که فرآیندی بی‌درنگ^۲ است از حجم محاسباتی کمی برخوردار باشد. به این ترتیب، میزان تأخیر محاسباتی سیستم (پارامتر T در شکل ۱۰) کم خواهد بود. که این مسأله تأثیر بسزایی در پایداری ربات و کنترل کننده آن دارد.

۴-۶ استخراج و شناسایی اشیا

در بخش قبل، با استفاده از فیلترهای منطبق رنگ زرد، قرمز و سیاه توانستیم تصویر صحنه پیش روی ربات را ناحیه‌بندی کنیم. حاصل الگوریتم ناحیه‌بندی برای هر فیلتر یک تصویر دو سطحی است که در آن پیکسل‌هایی که دارای رنگ مورد نظر هستند با یک و پیکسل‌های پس‌زمینه با صفر مشخص شده‌اند. در این زیربخش، هدف ما شناسایی اشیا در تصاویر دو سطحی مذکور است. فیلتر سیاه علاوه بر قوطی نوشابه خط هادی را هم آشکار می‌کند. فیلتر قرمز نیز علاوه بر جعبه صابون، برچسب قرمز رنگ قوطی نوشابه را نیز آشکار می‌کند. برای استخراج اشیا ابتدا تمام تصاویر دو سطحی (که از اعمال فیلترهای منطبق بر تصویر بوجود آمده‌اند) را با هم ترکیب می‌کنیم. سپس بر تصویر حاصل، عملگر ریخت‌شناسی بسته‌شدن^۳ اعمال می‌شود [۱۷]. به این ترتیب تکه‌های مختلف اشیا، به یکدیگر چسبیده، یکپارچه می‌گردند. از عملگرهای ریخت‌شناسی می‌توان برای حذف اشیا کوچک از تصویر نیز استفاده کرد [۳۷]. همانطور که می‌دانید خط هادی یک شی کشیده و باریک است؛ در حالی که بقیه اشیا موجود در تصویر متراکم و فشرده هستند. بنابراین با اعمال عملگر ریخت‌شناسی بازشدن^۴ بر روی تصویر ترکیبی، خط و اشیا مجازی ناشی از نویز حذف می‌شوند. پس از حذف اشیا مجازی، اکنون باید نوع اشیا باقی‌مانده مشخص شود. اینکار را بر اساس پاسخ‌های فیلترهای منطبق انجام می‌دهیم. اگر همه شی یا بخشی از آن توسط فیلتر منطبق رنگ سیاه آشکار شده باشد آن شی قوطی نوشابه بوده است. اگر شی فقط توسط یکی از فیلترهای زرد یا قرمز آشکار شده باشد، آنگاه آن شی به ترتیب توپ و یا جعبه صابون بوده است. چنانچه بخش‌های

^۱ Off-line

^۲ Real-time

^۳ Closing

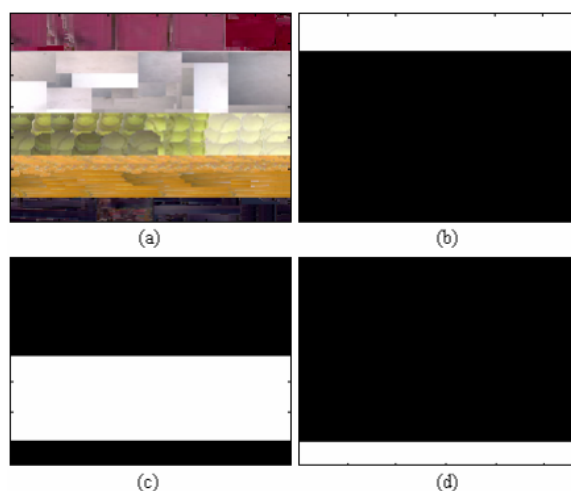
^۴ Opening

مختلف یک شی توسط هر سه فیلتر آشکار شده باشد، آنگاه آن شی، برجی است که از قرار گرفتن جعبه صابون، قوطی نوشابه و توپ بر روی هم ساخته می‌شود. بر اساس تعریف مسابقه، در زمین تنها یک برج قرار می‌گیرد و تشخیص آن از امتیاز اضافی برخوردار است. همانطور که مشاهده می‌شود، روش پیشنهادی حتی زمانی که شی مورد نظر مانند قوطی نوشابه دارای چندین رنگ باشد نیز بخوبی قادر به تشخیص آن است. بدیهی است که در این حالت، باید برای هر رنگ یک فیلتر منطبق رنگ مجزا در نظر گرفته شود. بدیهی است که پس از استخراج نواحی رنگی مختلف و ادغام آنها بروش فوق، می‌توان با دقت خوبی شی مورد نظر را از تصویر استخراج نمود.

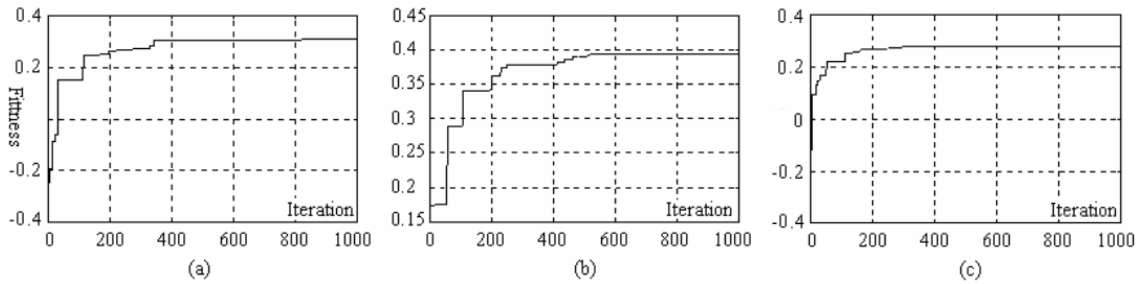
جعبه صابون و قوطی نوشابه در ابتدا به صورت عمودی در محوطه قرار داده می‌شوند. اما بسیار محتمل است که بر اثر برخورد ربات به آنها به پهلو بیافتند. به هر حال، روش پیشنهادی به شکل شی در تصویر وابسته نبوده و در صورت مشاهده شی با رنگ مورد نظر در تصویر قادر به تشخیص آن خواهد بود.

۷) نتایج تجربی

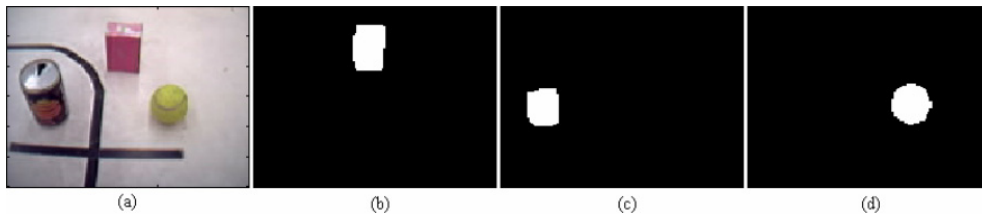
در شکل ۱۷، تصویر مرجعی که در این مقاله برای طراحی فیلترهای منطبق رنگ زرد، قرمز و سیاه مورد استفاده قرار گرفته، نشان داده شده است. همانطور که در زیربخش ۶-۱ اشاره گردید، این تصویر حاصل نمونه‌برداری از تصاویری است که تحت شرایط نوری مختلف تهیه شده‌اند. در این شکل، علاوه بر تصویر مرجع، نواحی مربوط به هر یک از سه رنگ فوق نیز نشان داده شده است. ما برای طراحی فیلترهای منطبق مذکور، از الگوریتم تکاملی پیشنهادی استفاده کرده‌ایم. در شکل ۱۸، منحنی تغییرات مقدار تابع هزینه بهترین کروموزوم جمعیت، در طول فرآیند تکامل GACFID نشان داده شده است. همانطور که مشاهده می‌کنید الگوریتم تکاملی پیشنهادی در تمام موارد طی کمتر از ۶۰۰ تولید نسل توانسته است به پاسخ بسیار مناسبی دست یابد. عملکرد فیلترهای طراحی شده برای سه تصویر نمونه در شکل‌های ۱۹-۲۱ نشان داده شده است. همانطور که مشاهده می‌شود، علیرغم تغییرات شدید شرایط نوری در تصاویر مذکور، الگوریتم پیشنهادی بخوبی توانسته است، اشیاء مورد نظر را بر اساس رنگ آنها شناسایی و استخراج نماید. شکل‌های ۱۹-b و ۲۰-b بیانگر آن هستند که با در نظر گرفتن شی مجازی در تصویر، الگوریتم ناحیه‌بندی بخوبی توانسته است، عدم حضور شی با رنگ قرمز در تصویر را تشخیص دهد. همچنین، اندازه‌گیری‌های تجربی نشان داد که فرآیند ناحیه‌بندی تصویر و شناسایی اشیاء (با استفاده از سه فیلتر منطبق) برای هر تصویر، طی مدت زمان متوسط ۰/۳ ثانیه (برای هر فیلتر ۰/۱ ثانیه) به انجام می‌رسد که این خود بیانگر حجم کم محاسبات روش پیشنهادی در مرحله بازرسی است. فیلترهای منطبق بر روی ربات متحرک نیز عملکرد قابل قبولی از خود به نمایش گذاشتند.



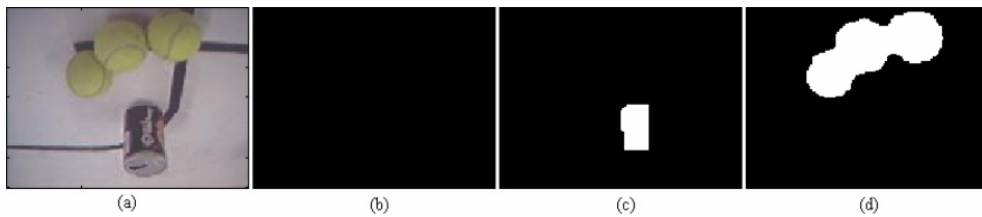
شکل ۱۷. (a) تصویر مرجعی که از نمونه‌برداری از تصاویر گرفته شده تحت شرایط نوری متفاوت بدست آمده است. در این مقاله، از این تصویر برای طراحی فیلترهای زرد، قرمز و سیاه استفاده شده است. در سایر تصاویر به ترتیب نواحی (b) قرمز رنگ، (c) زرد رنگ و (d) سیاه رنگ در تصویر مرجع نشان داده شده است.



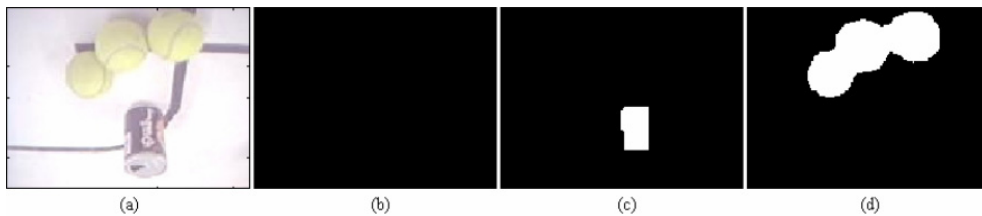
شکل ۱۸. منحنی‌های تغییرات مقدار تابع هزینه بهترین کروموزوم جمعیت، در طول فرآیند تکامل الگوریتم GACFID به منظور طراحی فیلترهای منطبق (a) قرمز، (b) زرد و (c) سیاه.



شکل ۱۹. عملکرد الگوریتم پیشنهادی در یک تصویر نمونه تحت شرایط نوری متغیر. (a) اشیا قرمز رنگ، (b) اشیا سیاه رنگ و (c) اشیا زرد رنگ آشکار شده در تصویر نمونه.



شکل ۲۰. عملکرد الگوریتم پیشنهادی در یک تصویر نمونه با نور یکنواخت. (a) اشیا قرمز رنگ، (b) اشیا سیاه رنگ و (c) اشیا زرد رنگ آشکار شده در تصویر نمونه.



شکل ۲۱. عملکرد الگوریتم پیشنهادی در یک تصویر نمونه تحت نور شدید. (a) اشیا قرمز رنگ، (b) اشیا سیاه رنگ و (c) اشیا زرد رنگ آشکار شده در تصویر نمونه.

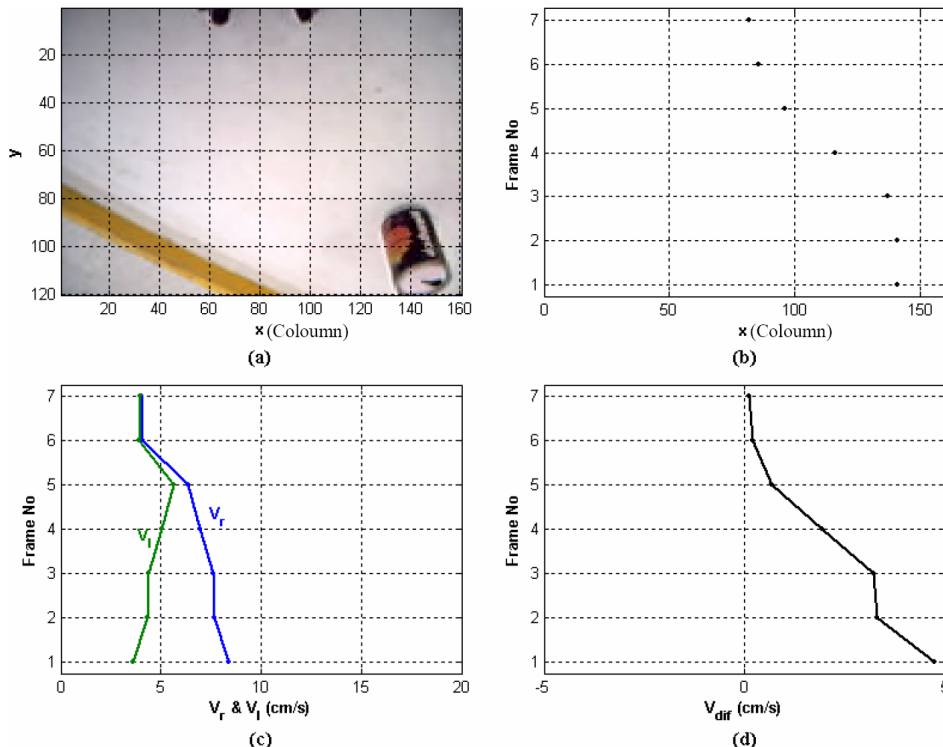
بعلاوه، سیستم هدایت ترکیبی ربات، به آن انعطاف‌پذیری زیادی بخشیده است. بطوری که قادر است با بکارگیری رفتارهای پایه، تصمیم‌گیری‌های پیچیده‌ای چون شکار اشیا موجود در زمین را انجام دهد. کنترل‌کننده دیداری طراحی شده برای ربات نیز از پایداری زیادی برخوردار است. بطوری که قادر است در شرایط مختلف، ربات را بسوی هدف تعیین شده هدایت کند. در شکل‌های ۲۲-۲۴ عملکرد سیستم هدایت و کنترل‌کننده ربات برای سه شی، به ترتیب قوطی نوشابه (در سمت چپ ربات)، جعبه صابون (در مقابل ربات) و توپ (در سمت راست ربات) نشان داده شده است. در هر یک از شکل‌های فوق، بخش (a) اولین فریمی را که شی در آن ظاهر گردیده است، نشان می‌دهد. در بخش‌های بعدی نیز به ترتیب، تغییرات موقعیت شی در تصویر، سرعت چرخ‌های ربات و تفاضل سرعت‌ها ($V_{dif} = V_r - V_l$) طی فریم‌های بعدی نشان داده شده است. به وضوح می‌توان مشاهده نمود که کنترل‌کننده دیداری به نرمی و با دقت قابل قبولی ربات را به سوی هدف هدایت نموده است؛ بطوری که پس از حداکثر ۱۰ فریم، شی در درون بازوی ربات قرار

گرفته است. این درحالی است که در این ربات، دوربین واسنجی نشده و خطای زاویه و فاصله ربات صرفاً از روی تصاویر دریافتی تخمین زده می‌شود (بخش ۵-۱ را ببینید).

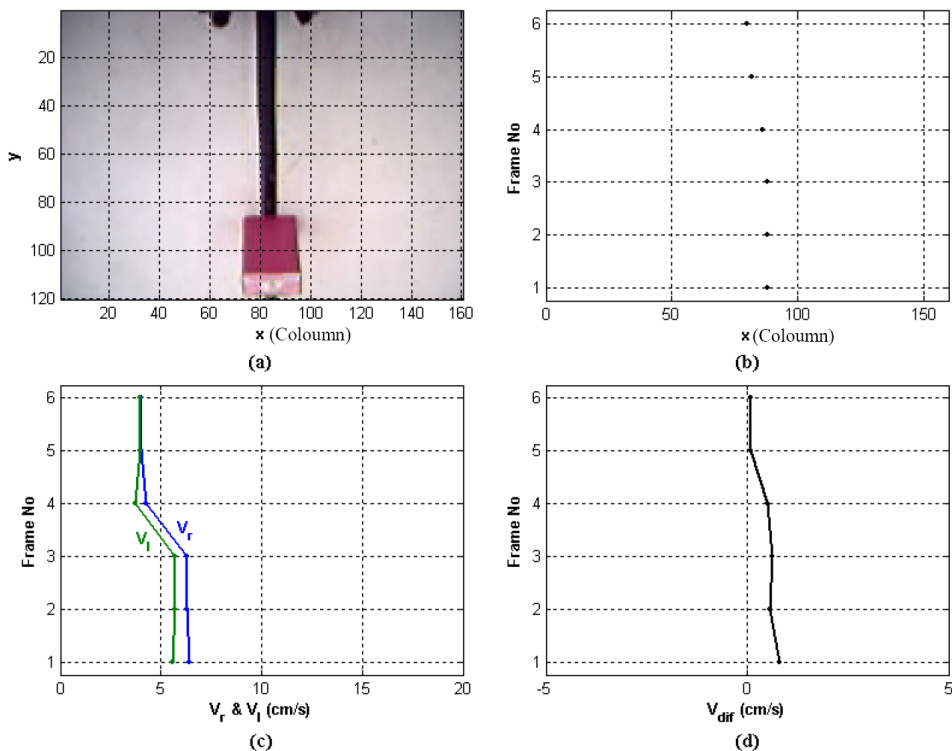
در شرایط آزمایشگاهی، ربات بارها به آسانی توانست، تحت شرایط مختلف محیطی، تمام اشیاء موجود در زمین را شکار و به دروازه مربوطه منتقل نماید. همچنین، این ربات در دومین دوره مسابقات سراسری رباتیک دانشگاه تهران شرکت نموده، توانست ظرف مدت تنها چهار دقیقه (تحت شرایط نوری بسیار نامناسب) یک قوطی نوشابه، یک جعبه صابون و دو توپ را شکار کند.

۸ نتیجه‌گیری

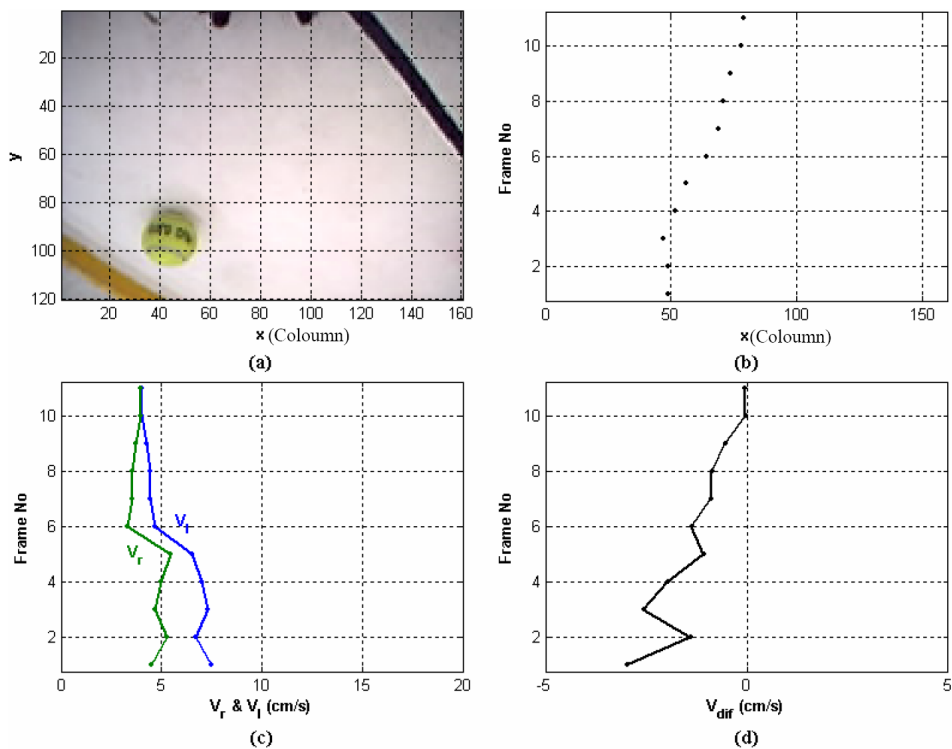
در این مقاله، یک ربات متحرک بینا معرفی شده است. در سیستم هدایت این ربات که بر پایه معماری ترکیبی بنیان نهاده شده، از پنج رفتار واکنشی و سه رفتار هدف‌گرا استفاده گردیده است. رفتارهای هدف‌گرای ربات همگی براساس بینایی ماشینی شکل گرفته‌اند. هر سه رفتار هدف‌گرای مذکور از یک کنترل کننده تناسبی-مشتق‌گیر-تناسبی با پس‌خورد تصویری، برای کنترل ربات استفاده می‌کنند. در این ربات، برای شناسایی اشیاء از ویژگی رنگ آنها استفاده شده است. بر این اساس، در این مقاله، روشی جدید برای آشکارسازی رنگ تحت عنوان فیلتر منطبق رنگ ارائه شده است. به منظور طراحی بهینه این فیلترها، از الگوریتم‌های ژنتیکی استفاده گردیده است. نتایج تجربی بیانگر عملکرد چشمگیر فیلترهای منطبق رنگ، تحت شرایط نوری مختلف و حجم کم محاسباتی آنها در مرحله بازرسی است. سیستم هدایت ترکیبی ربات نیز عملکرد رضایت‌بخشی از خود نشان می‌دهد؛ بطوری که ربات قادر است تحت شرایط محیطی مختلف، تمام اشیاء موجود در زمین مسابقه را شکار و به دروازه‌های مربوطه ترابری کند. کنترل‌کننده دیداری، قابلیت اطمینان مناسبی به ربات بخشیده است؛ به طوری که در آزمایشات تجربی متعدد، هیچگاه مشاهده نگردید که ربات در حرکت به سوی هدف از پیش تعیین شده، دچار خطا گردیده و یا در شکار شی ناموفق باشد. در مجموع می‌توان گفت که ربات طراحی شده، یک نمونه موفق از ربات‌هایی است که از معماری ترکیبی برای هدایت آنها استفاده شده است.



شکل ۲۲. عملکرد کنترل‌کننده دیداری برای یک قوطی نوشابه در سمت چپ ربات. (a) اولین فریمی که شی در آن دیده شده است. (b) تغییرات موقعیت شی در تصویر، (c) تغییرات سرعت چرخ‌های ربات و (d) تغییرات اختلاف سرعت دو چرخ، از زمان مشاهده شی برای اولین بار تا هنگامی که شی شکار می‌شود.



شکل ۲۳. عملکرد کنترل کننده دیداری برای یک جعبه صابون در مقابل ربات. (a) اولین فریمی که شی در آن دیده شده است. (b) تغییرات موقعیت شی در تصویر، (c) تغییرات سرعت چرخ‌های ربات و (d) تغییرات اختلاف سرعت دو چرخ، از زمان مشاهده شی برای اولین بار تا هنگامی که شی شکار می‌شود.



شکل ۲۴. عملکرد کنترل کننده دیداری برای یک توپ در سمت راست ربات. (a) اولین فریمی که شی در آن دیده شده است. (b) تغییرات موقعیت شی در تصویر، (c) تغییرات سرعت چرخ‌های ربات و (d) تغییرات اختلاف سرعت دو چرخ، از زمان مشاهده شی برای اولین بار تا هنگامی که شی شکار می‌شود.

در پایان بر خود لازم می‌دانیم از زحمات اساتید، مسؤولان و کارمندان دانشکده مهندسی دانشگاه فردوسی مشهد که همواره یار و مشوق ما بودند تقدیر و تشکر نماییم.

منابع

1. Egerton, S. J., and Callaghan, V., "From mammals to machines: towards a biologically inspired mapping model for autonomous mobile robots," *6th Int. Conf. on Intelligent Autonomous Systems*, (2000).
2. Fukuda, T., Michelini, R., Potkonjak, V., Tzafestas, S., Valavanis, K., and Vukobratovic, M., "How far away is artificial man," *IEEE Robotics & Automation Magazine*, 8(1), pp. 66–73, (2001).
3. Titterton, D. H., and Weston, J. L., "Strapdown Inertial Navigation Technology", 2nd ed., IEE, Peter Reregrinus Ltd., (1991).
4. Murphy, R. R., "Introduction to AI Robotics", 1st ed., MIT Press, (2000).
5. Kortenkamp, D., Bonasso, R. P., and Murphy, R., "Review of Artificial Intelligence and Mobile Robotics", AAAI Press/MIT Press, (1998).
6. Gupta, M., and Sinha, N., "Intelligent Control Systems Theory and Applications", IEEE Press, (1996).
7. Arkin, R. C., "Behavior Based Robotics", MIT Press, (1998).
8. Dudek, G., and Jenkin, M., "Computational Principles of Mobile Robotics", Cambridge University Press, (2000).
9. Tunstel, E., and Jamshidi, M., "Behavior hierarchy for autonomous mobile robots: fuzzy behavior modulation and evolution," *Intelligent Automation and Soft Computing*, 3(1), pp. 37–50, (1997).
10. McFetridge, L., and Ibrahim, M. Y., "A new methodology of mobile robot navigation: the agoraphilic algorithm," *Robotics and Computer-Integrated Manufacturing*, (2008), doi:10.1016/j.rcim.2008.01.008. (Article in Press)
11. Luh, G.-C., and Liu, W.-W., "An immunological approach to mobile robot reactive navigation," *Applied Soft Computing*, 8, pp. 30–45, (2008).
12. Huq, R., Mann, G. K. I., and Gosine, R. G., "Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation," *Applied Soft Computing*, 8, pp. 422–436, (2008).
13. Labrosse, F., "Short and long-range visual navigation using warped panoramic images," *Robotics and Autonomous Systems*, 55, pp. 675–684, (2007).
14. Hayet, J. B., Lerasle, F., and Devy, M., "A visual landmark framework for mobile robot navigation," *Image and Vision Computing*, 25, pp. 1341–1351, (2007).
15. Huang, W. H., Fajenb, B. R., Finka, J. R., and Warrenc, W. H., "Visual navigation and obstacle avoidance using a steering potential function," *Robotics and Autonomous Systems*, 54, pp. 288–299, (2006).
16. Fernandez-Leon, J. A., Acosta, G. G., and Mayosky, M. A., "Behavioral control through evolutionary neurocontrollers for autonomous mobile robot navigation," *Robotics and Autonomous Systems*, (2008), doi:10.1016/j.robot.2008.06.012. (Accepted Manuscript)
17. Gonzalez, R. C., and Woods, R. E., "Digital Image Processing", 2nd ed., Prentice-Hall, (2003).
18. Golub, G. H., and Loan, C. F. V., "Matrix Computations", Johns Hopkins University Press, Baltimore, pp. 557–558, (1989).
19. Chen, H.-C., Chien, W.-J., and Wang, S.-J., "Contrast-based color image segmentation," *IEEE Signal Processing Letters*, 11(7), pp. 641–644, (2004).
20. Shafarenko, L., Petrou, M., and Kittler, J., "Histogram-based segmentation in a perceptually uniform color space," *IEEE Trans. Image Processing*, 7(9), pp. 1354–1358, (1998).
21. Shafarenko, L., Petrou, M., and Kittler, J., "Automatic watershed segmentation of randomly textured color images," *IEEE Trans. Image Processing*, 6(11), pp. 1530–1543, (1997).
22. Gevers, T., "Image segmentation and similarity of color-texture objects," 4(4), pp. 509–516, (2002).
23. Yang, L., Meer, P., and Foran, D. J., "Unsupervised segmentation based on robust estimation and color active contour models," *IEEE Trans. Information Technology in Biomedicine*, 9(3), pp. 475–485, (2005).
24. Drew, M. S., Wei, J., and Li, Z. N., "Illumination on-invariant color object recognition via compressed chromatic histograms of color channel normalized images," *IEEE Int. Conf. on Computer Vision*, pp. 533–540, (1998).
25. Netraval, A. N., and Haskell, B. G., "Digital Pictures, Representation and Compression", 2nd printing, Plenum Press, AT&T Bell Laboratories, (1989).
26. Bujesy, R., and Lee, S. W., "Color image segmentation with detection and illumination induced by inter reflection," *IEEE Int. Conf. on Pattern Recognition*, 1, pp. 785–790, (1990).
27. Eum, K. B., and Lee, J., "Color image segmentation using positivistic approach," *IEEE Int. Conf. on Systems, Man and Cybernetics*, 2, pp. 1150–1155, (1996).

28. Moghaddamzadeh, A., and Bourbakis, N., "A fuzzy technique for image segmentation of color images," *IEEE World Cong. on Computational Intelligence*, 1, pp. 83–88, (1994).
29. Whitley, D., "A genetic algorithm tutorial," *Statistics and Computer*, 4, pp. 65–85, (1994).
30. Holland, J., "*Adaptation in Natural and Artificial Systems*", University of Michigan Press, (1975).
31. Samii, R., "*Electromagnetic optimization by genetic algorithms*", Wiley-RsP, (1999).
32. Darwin, C., "*The Origin of Species by Means of Natural Selection*", Murray, (1859).
33. Whitley, D., "The GENITOR Algorithm and Selective Pressure", *3rd Int. Conf. on Genetic Algorithms*, ed. Morgan-Kaufmann: pp. 116–121, (1989).
34. Goldberg, D. E., "A Note on Boltzman Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing", *Complex Systems*, 4, pp. 445–60, (1990).
35. Rudolph, G., "Convergence Properties of Canonical Genetic Algorithms", *IEEE Trans. Neural Networks*, NN-5, pp. 96–101, (1994).
36. Parker, J. R., "*Algorithms for Image Processing and Computer Vision*", John Wiley & Sons, (1997).
37. Sonka, M., Hlavac, V., and Boyle, R., "*Image Processing Analysis and Machine Vision*", International Thomson Computer Press, (1996).