



ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

## تقسیم بندی سریع و کم هزینه تصاویر رنگی برای ربات های تعاملی

### چکیده

سیستم های دیداری که از تقسیم منطقه به رنگ استفاده می کنند در برنامه های ربات موبایل بلادرنگ مانند RoboCup یا سایر زمینه هایی که تعامل با انسان یا جهان پویا نیاز باشد به کار می روند. به طور سنتی سیستم هایی که قطعه بندی بلادرنگ رنگی را به کار می برند یا در سخت افزار پیاده سازی شده اند یا به عنوان سیستم های نرم افزاری بسیار خاص از مزایای سایر زمینه های علمی برای رسیدن به کارایی ضروری استفاده می شوند. با این حال متوجه شدیم که با دقت به کارایی الگوریتم، قطعه بندی سریع تصویر رنگی با استفاده از قطعه تصاویر گرفته شده و سخت افزار CPU قابل انجام شدن است. مقاله ما سیستمی توصیف میکند که قادر به دنبال کردن چند صد ناحیه از ۳۲ رنگ با فرکانس ۳۰ هرتز در قطعه های همه منظوره سخت افزاری می باشد. سیستم سخت افزاری از ۴ قسمت اصلی تشکیل شده است: یک پیاده سازی جدید از یک **threshold classifier**، یک سیستم ادغام به منظور ایجاد قسمت هایی از طریق اجزای منفصل ایجاد می شود، یک سیستم جدا کننده و مرتب کننده و یک ادغام کننده بالا به پایین ابتکاری به منظور گروه بندی ادراکی تقریبی که امکانات مختلف نواحی را گردآوری میکند. کلید کارایی دیدگاه ما، روش جدیدی برای به انجام رساندن آستانه فضای رنگی است که یک پیکسل را قادر می سازد تا در یک تا ۳۲ رنگ با استفاده از تنها دو عملگر **AND** منطقی، طبقه بندی شود. یک رویکرد ساده می تواند نیازمند ۱۹۲ مقایسه برای طبقه بندی مشابه باشد. الگوریتم و نمایه ها توصیف شده اند و همچنین ۳ کاربرد استفاده شده نیز توصیف شده است.

### ۱ مقدمه

اولین گام ابتدایی مهم در بسیاری از اعمال دیداری رنگی، طبقه بندی هر پیکسل در یک تصویر به تعدادی کلاس های رنگی مجزا است. رویکرد مورد استفاده در به انجام رساندن این عمل شامل آستانه رنگ های خطی (**Linear Color Thresholding**)، طبقه بندی نزدیک ترین همسایه، آستانه فضای رنگی و روش های احتمالی می باشد.

آستانه رنگ های خطی با پارتیشن بندی ناحیه رنگی با مرزهای خطی کار می کند(به عنوان مثال فضای سه بعدی). یک پیکسل به خصوص براساس اینکه در کدام پارتیشن قرار گرفته طبقه بندی می شود. این روش برای یادگیری سیستم هایی مانند شبکه های عصبی یا درختهای تصمیم گیری چند متغیره مناسب است.

رویکرد دوم استفاده از نزدیک ترین طبقه بندی همسایه است. معمولا چند صد نمونه قبل از طبقه بندی به کار گرفته می شوند، هر کدام موقعیت منحصر به فردی در فضای رنگ و طبقه بندی مرتبط دارند. برای طبقه بندی یک پیکسل جدید، لیستی از K تا از نزدیکترین نمونه ها یافت میشود، سپس پیکسل براساس بزرگترین قسمت از طبقه بندی همسایگان طبقه بندی می شود. هر دو آستانه خطی و نزدیکترین طبقه بندی همسایه نتایج خوبی از لحاظ دقت طبقه بندی فراهم می کنند، اما عملکرد بلادرنگ با استفاده از سخت افزار در دسترسی فراهم نمی کنند.

رویکرد دیگر، استفاده از مجموعه ای از آستانه های ثابت است که کلاس های رنگ را به عنوان یک بلاک مستطیل شکل در فضای رنگ تعریف میکنند. این رویکرد عملکرد خوبی را فراهم میکند اما قادر به استفاده از مزایای وابستگی بین ابعاد فضای رنگی نمی باشد. نوع دیگری از آستانه ثابت در سخت افزار توسط Newton Laboratories پیاده سازی شده است. محصول آنها، ردیابی داده های رنگی به صورت بلادرنگ را فراهم میکند اما گرانتر از رویکردهای صرفا نرم افزاری است.

آخرین رویکرد مرتبط ذخیره نسخه گسسته ای از تمام توزیع احتمال های مشترک است. در این روش به منظور بررسی اینکه آیا یک پیکسل خاص عضوی از کلاس رنگ هست یا نه، اجزای فردی رنگ به عنوان شاخص هایی برای نمودار ستونی چند بعدی استفاده می شود. هنگامی که موقعیت تعیین شد، شماره احتمال حقیقی عضویت در برگشت فرستاده می شود. این روش مدلسازی حجمهای با توزیع دلخواه را فراهم می کند و عضویت با کارایی معقول می تواند بررسی شود. این رویکرد همچنین کاربر را قادر می سازد تا حجم های عضویت نامعمول را نمایش دهد(مانند مخروط یا بیضی) و بنابراین وابستگی بین ابعاد فضای رنگی را میگیرد. اشکال اصلی این رویکرد هزینه زیاد حافظه برای افزایش سرعت احتمال ماتریسی است که باید در حافظه قرار گیرد.

رویکردی که در کارمان استفاده می کنیم، ترکیبی از روش های توصیف شده در بالا است اما با تمرکز مخصوص بر مسائل بهره وری. بنابراین میتوانیم طبقه بندی موثر در نرخ های بلادرنگ را فراهم کنیم. این روش به عنوان بهترین آستانه ثابت تعریف شده است، اما با یک فضای رنگی برنامه ریزی شده در هنگام نیاز. بالای این لایه ای است که قالب را به نمایش های هندسی تر مناسب برای پردازش های سطح بالا تبدیل می کند. در بخش بعدی طرح کلی رویکرد ما ارائه می شود. بخش های باقیمانده عملکرد سیستمی که از این روش استفاده می کند را توصیف میکند و مثالهایی از چندین کاربرد آن را ارائه می دهد.

## ۲ توصیف رویکرد

### ۲.۱ تغییر شکل فضای رنگی

رویکرد ما شامل استفاده از آستانه ها (thresholds) در فضای رنگی سه بعدی است. فضا های با چندین رنگ به طور گسترده استفاده می شوند، شامل (Hue Saturation Intensity) HSI، (Red Green Blue) RGB و YUV. انتخاب فضای رنگی برای طبقه بندی به چندین عامل بستگی دارد که از جمله آن ها می توان به دیجیتالی شدن سخت افزار و مفید بودن برای کاربرد مشخصی اشاره کرد.

RGB فضای رنگی شناخته شده ای است که اغلب در پردازش تصویر استفاده می شود، اما در بسیاری از کاربردهای دیداری رباتیک دچار تغییر می شود. مثلا فوتبال رباتیک را در نظر بگیرید، که در آن ویژگی های محیط با شناسایی رنگ ها نشانه گذاری شده اند (مثلا توپ ممکن است نارنجی باشد). ما تمایل داریم نرم افزار طبقه بندی ما در روشنایی های مختلف قدرتمند عمل کند، پس تعریف رنگ نارنجی با نسبتی از سبز و قرمز و آبی در پیکسل می تواند مفید باشد. این امر در فضای رنگی RGB امکان پذیر است اما حجم ضمنی توسط این رابطه مخروطی است و توسط آستانه های ساده قابل ارائه نمی باشد.

در contrast، HSI و YUV مزایایی دارند که رنگ را به کدهایی در دو بعد تبدیل میکنند (H و S برای HSI و U و V برای YUV) در حالیکه شدت رنگ به صورت سه بعدی کد گذاری می شود. این فضاهای رنگی از این جهت از RGB برای استفاده های رباتیک مفیدتر هستند.

بعضی از سخت افزار های دیجیتالی یک یا تعداد بیشتری فضاهای رنگی مناسب را به طور مستقیم فراهم می کنند. در سایر موارد، ممکن است فضا نیازمند تغییر فضای فراهم شده توسط سخت افزار به چیز مناسب تری باشد. هنگامی که یک طرح مناسب انتخاب شد، از آنجا که بیشتر رابطه های قابل توجه حذف شده اند، فضای نتیجه ممکن است با استفاده از آستانه های ثابت پارتیشن بندی شود

قطعه دیجیتالی که در ابتدا استفاده کردیم، تصاویر با کدگذاری RGB تولید می کند. ما متوجه شدیم که rotate کردن فضای رنگی RGB به طرز قابل توجهی ردیابی را قدرتمند تر می کند. بیشتر اطلاعات در یک تصویر RGB، در امتداد محور intensity (شدت) متفاوت هستند که تقریباً اشعه محور سه رنگ را به دو نیم تقسیم می کند. با محاسبه شدت و کم کردن این قسمت از هر مقدار رنگ، فضایی که در آن واریانس قرار می گیرد به موازات محور ایجاد می شود، که ناحیه را به صورت دقیق تری به شکل یک مستطیل نمایش می دهد.

یکی دیگر از تغییر شکل های قدرتمند اما پر هزینه، استفاده از فضای رنگ RGB کسری است که هر رنگ قسمت ها به عنوان کسری از شدت مشخص می شود و شدت به عنوان بعد دیگری اضافه می شود. این موضوع باعث ایجاد طرح ۴ بعدی با فضای ثابت دقیق می شود اما با وجود بعد اضافی برای پردازش و ۳ تقسیم هر پیکسل برای محاسبه کسر، اثبات می کند که برای سخت افزار موجود در حال حاضر کندتر است.

بعد به سیستمی منتقل شدیم که رنگ های YUV را در سخت افزار فراهم می کرد. این موضوع قدرت یک فضای قوی را بدون خطای عملکرد یک نرم افزار تغییر رنگ ترکیب می کرد. بنابراین سیستم ها می توانند از سخت افزار با فضاهای رنگی بومی استفاده کنند اما حتی بدون آنها، یک تغییر مناسب می تواند منجر به راه حل معقولی شود.

## ۲،۲ Thresholding (آستانه)

روش آستانه که در اینجا توضیح داده شده است می تواند با فضاهای رنگی چند بعدی که اجزای سطوح رنگی گسسته دارند استفاده شود، اما برای مثال در رویکردمان از فضای رنگی YUV استفاده خواهیم کرد، هر کلاس رنگ به عنوان مجموعه ای از ۶ مقدار آستانه مشخص شده است: دو تا برای هر بعد در فضای رنگی، پس از تغییر اگر یکی مورد استفاده قرار گرفته باشد. مکانیزم استفاده شده برای آستانه یکی از نکات مهم بهره وری است زیرا عملیات آستانه باید

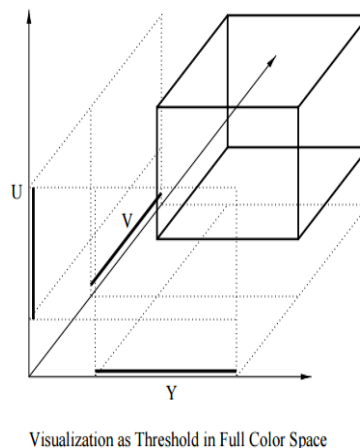
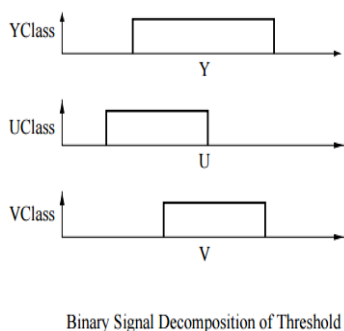
برای هم رنگ در هر پیکسل تصویر تکرار شود. یک راه برای بررسی اینکه آیا یک پیکسل عضوی از یک کلاس رنگ به خصوصی هست یا نه این است که از مجموعه ای از مقایسه های استفاده کنیم تا مشخص شود که آیا پیکسلی با مقادیر  $Y$  و  $U$  و  $V$  باید در کلاس رنگ گروه بندی شود یا نه.

```

if ((Y >= Ylowerthresh)
    AND (Y <= Yupperthresh)
    AND (U >= Ulowerthresh)
    AND (U <= Uupperthresh)
    AND (V >= Vlowerthresh)
    AND (V <= Vupperthresh))
    pixel_color = color_class;

```

متأسفانه این رویکرد نسبتاً نامناسب است زیرا، هنگامی که کامپایل می شود، ممکن است به ۶ شاخه شرطی نیاز داشته باشد تا عضویت یک پیکسل در کلاس رنگ را مشخص کند. این امر مخصوصاً در پردازش های خط لوله (pipelined) با اجرای دستورالعمل های مخاطره آمیز نامناسب باشد.



شکل ۱ : ناحیه سه بعدی فضای رنگی برای طبقه بندی به عنوان ترکیبی از ۳ تابع باینتری نمایش داده شده است. در عوض پیاده سازی ما از یک مقدار Boolean از آستانه های چند بعدی استفاده می کند. ناحیه های این چینی می توانند به عنوان محصول ۳ تابع نمایش داده شوند، در هر یک از محور های فضای (شکل ۱). تجزیه نشان داده شده

برای هر مقدار از یک جزء رنگ در آرایه ها ذخیره می شود. بنابراین عضویت در کلاس می تواند به عنوان AND بیتی از عناصر هریک از آرایه های نشان داده شده با ارزش جزء رنگ محاسبه شود:

```
pixel_in_class = YClass[Y]
                AND UClass[U]
                AND VClass[V];
```

مقدار بولی حاصل از pixel\_in\_class نشان می دهد که آیا پیکسل متعلق به کلاس است یا نه. این روش به سیستم اجازه می دهد تا به صورت خطی تعداد پیکسل ها و ابعاد فضای رنگ را تنظیم کند و به عنوان چند آرایه برای هر پیکسل در نظر گرفته شود. عملیات بسیار سریعتر از روش ساده است زیرا AND بیتی به طرز قابل توجهی نسبت به مقایسه integer در بیشتر پردازنده های مدرن هزینه را کاهش می دهد.

برای نشان دادن رویکرد، مثال زیر را در نظر بگیرید. فرض کنید ما فضای رنگ YUV را به ۱۰ سطح در هر بعد تقسیم کرده ایم. بنابراین نارنجی به عنوان مثال، ممکن است با اختصاص مقادیر زیر به عناصر هریک از آرایه نشان داده شود:

```
YClass[] = {0,1,1,1,1,1,1,1,1,1};
UClass[] = {0,0,0,0,0,0,0,1,1,1};
VClass[] = {0,0,0,0,0,0,0,1,1,1};
```

بنابراین برای بررسی اینکه آیا یک پیکسل با مقدار رنگ (1,8,9) عضوی از کلاس رنگ نارنجی است یا نه، تنها کاری که باید انجام دهیم بررسی عبارت YClass[1] AND UClass[8] AND VClass[9] است که در این مورد مقدار ۱ یا true را نتیجه می دهد که نشان دهنده عضو بودن رنگ در کلاس نارنجی است.

یکی از مهمترین مزایای رویکرد ما این است که می توان عضویت یک پیکسل در چند کلاس را به طور همزمان مشخص کرد. با به کار گیری parallelism در عملیات بیتی AND برای اعداد، می توانیم عضویت چندین کلاس را همزمان تشخیص دهیم. به عنوان مثال، فرض کنید منطقه ای از فضای رنگ توسط پیکسل های آبی که ارائه شده به شرح زیر اشغال شده است:

```
YClass[] = {0,1,1,1,1,1,1,1,1,1};  
UClass[] = {1,1,1,0,0,0,0,0,0,0};  
VClass[] = {0,0,0,1,1,1,0,0,0,0};
```

به جای ساخت یک مجموعه جداگانه از آرایه برای هر رنگ، ما می توانیم با استفاده از موقعیت هر بیت در یک عنصر آرایه آن ها را با هم ترکیب کنیم که نشان دهنده مقادیر متناظر برای هر رنگ می باشد، بنابراین، به عنوان مثال، اگر هر عنصر در یک آرایه یک عدد صحیح دو بیتی بود ما می توانیم نمایش های آبی و نارنجی را به صورت زیر ترکیب کنیم:

```
YClass[] = {00,11,11,11,11,11,11,11,11,11};  
UClass[] = {01,01,01,00,00,00,00,10,10,10};  
VClass[] = {00,00,00,01,01,01,00,10,10,10};
```

که در آن اولین(اولویت بالا) بیت در هر عنصر برای نشان دادن نارنجی و دومین بیت به منظور نمایش آبی استفاده شده است. بنابراین می توانیم با ارزیابی عبارت `YClass[1] AND UClass[8] AND VClass[9]` بررسی کنیم آیا (1,8,9) یکی از دو کلاس هست یا نه. نتیجه ۱۰ می باشد که نشان می دهد رنگ در کلاس نارنجی است اما در کلاس آبی نیست.

در پیاده سازی ما، هر آرایه یک عدد صحیح ۳۲ بیتی است. بنابراین می توانیم ارزیابی عضویت در ۳۲ کلاس رنگ متمایز را با عملگر AND به یکباره انجام دهیم. در مقابل، رویکرد مقایسه ساده ممکن است ۳۲ \* ۶ تا ۱۹۲ مقایسه برای همان عملیات نیاز داشته باشد. علاوه بر این با توجه با اندازه کوچک کلاس رنگ نماینده، الگوریتم می تواند از اثرات ذخیره شده حافظه استفاده کند.

## ۲,۳ مناطق متصل

پس از اینکه نمونه رنگ های مختلف طبقه بندی شدند، مناطق متصل با بررسی نمونه های طبقه بندی شده تشکیل می شوند. این به طور معمول عملیاتی پر هزینه است که به شدت می تواند بر عملکرد عملیات بلادرنگ تاثیر گذارد. اجزای متصل ما رویه های پیاده سازی شده را در دو مرحله بنا به دلایل بازدهی ادغام می کنند.



مرحله اول محاسبه طول اجرا کدگذاری نسخه برای طبقه بندی تصویر است. در بسیاری از برنامه های کاربردی بینایی رباتیک تغییرات قابل توجه در پیکسل های مجاور تصویر نسبتا نادرنند. با گروه بندی پیکسل های مجاور مشابه به عنوان یک run تنها یک فرصت برای بهره وری داریم زیرا کاربران بعدی داده میتوانند در تمام run (اجرا)ها به جای پیکسل های فرد عمل کنند. هم چنین منافع عملی وجود دارد که ادغام مناطق تنها نیاز به اتصال عمودی دارد، زیرا اجزای افقی در تغییر به تصویر RLE ادغام شده اند.

روش ادغام از الحاق مبتنی بر درخت برای پیدا کردن مسیر فشرده سازی استفاده می کند. این موضوع عملکردی را ارائه می دهد که نه تنها در عمل خوب است بلکه همچنین یک الگوریتم سخت محدود فراهم می کند که برای تمام اهداف عملی خطی مورد استفاده قرار می گیرد. ادغام در محل بر روی تصویر RLE طبقه بندی شده انجام می شود. دلیل این است که هر run شامل فیلدی با تمام اطلاعات ضروری می باشد، یک شناسه عنصر والد یک run را نشان می دهد (سمت چپ ترین عنصر بالای منطقه). در ابتدا هر run خود را به عنوان والدش برچسب گذاری می کند، در نتیجه جنگل کاملا از هم گسسته ای حاصل می شود. اسکن های رویه ادغام ردیف ها را همجوار می کند و run هایی که از یک کلاس رنگ هستند و تحت ارتباط ۴ همپوشانی هستند را ادغام می کند. این یک جنگل گسسته که اشاره گر هر run والد به سمت بالا طرف والد عمومی منطقه اشاره می کند را نتیجه می دهد. بنابراین یک گذر یک ثانیه ای لازم است تا تمام مسیر ها را فشرده کند تا هر run با والد واقعی خودش برچسب گذاری شود. حال هر مجموعه run به یک والد منحصر به فرد اشاره می کند که یک منطقه متصل را مشخص می کند. پردازش در شکل ۲ نشان داده شده است.

#### ۲,۴ استخراج اطلاعات منطقه ای

در مرحله بعد اطلاعات منطقه ای را از نقشه RLE ادغام شده استخراج می کنیم. جعبه محدود شده در مرکز قرار می گیرد و اندازه منطقه به طور تدریجی در یک pass روی اولین ساختار داده جنگل محاسبه می شود. زیرا الگوریتم در حال pass یک run روی تصویر است نه پردازش یک منطقه، برچسب های منطقه دوباره شماره گذاری می شوند تا

هر برچسب منطقه شاخصی از یک ساختار منطقه در جدول منطقه باشد. این امر مراجعه را به طور قابل توجهی سریعتر می کند در حالیکه آمار افزایشی را محاسبه می کند.

سایر آمار می توانند به صورت اختیاری با pass های اضافی محاسبه شوند. در حال حاضر تنها گزینه اضافی محاسبه میانگین رنگ منطقه است که می تواند به عنوان یک آستانه دو منظوره برای بررسی اینکه اگر یک منطقه همان رنگ مطلوب object را دارد یا نه استفاده شود. سایر امکاناتی که ممکن است در آینده اضافه شوند عبارتند از واریانس رنگ در یک منطقه و پوسته های محدب و نقاط لبه که می تواند برای مدل هندسی اتصالات مفید واقع شود. پس از محاسبه آمار، مناطق بر اساس رنگ به موضوع های مختلف در لیست های پیوندی در جدول مناطق جدا می شوند. در نهایت آنها براساس اندازه مرتب می شوند تا الگوریتم های پردازش سطح بالا بتوانند با مناطق بزرگتر سر و کار داشته باشند و از مناطق نسبتاً کوچک چشم پوشی کنند.

## ۲.۵ ادغام مناطق مبتنی بر تراکم

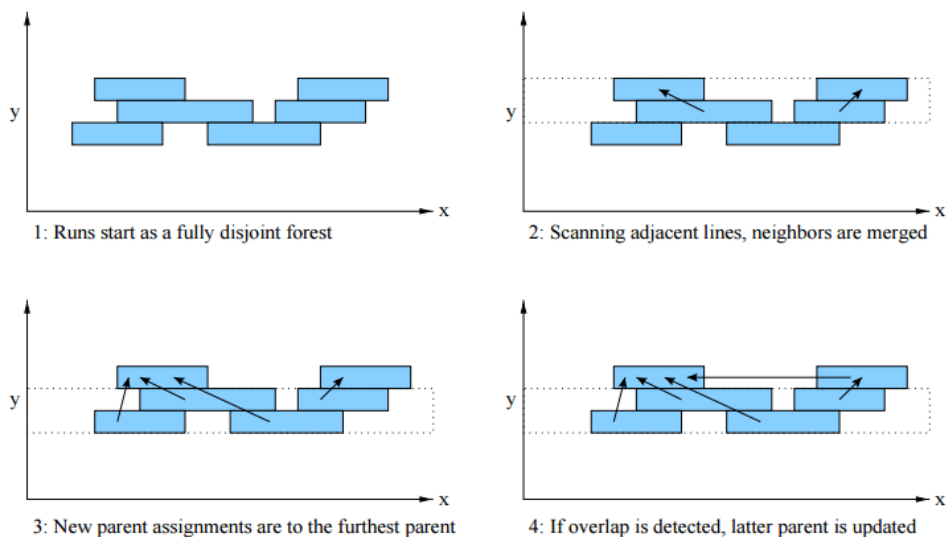
در لایه آخر قبل از اینکه داده به لایه بالایی به کاربر منتقل شود، یک ادغام ابتکاری بالا به پایین انجام می شود که به حذف بعضی از خطاهای تولیدی در پایین به بالا منطقه تولید شده کمک می کند. مشکل آدرس دهی شده در اینجا بهتر است با یک مثال معرفی شود. اگر یک منطقه شناسایی شده دارای یک خط از پیکسل های به اشتباه تشخیص داده شده بود، سطوح پایین تر از سیستم دیداری ممکن است آن را به عنوان دو منطقه مجزا تشخیص دهند تا یک منطقه. بنابراین یک تغییر کوچک در ورودی اول می تواند نتایج بسیار متفاوتی داشته باشد.

یک راه حل به کارگیری یک نوع گروه بندی ابتکاری است که در آن object ها مشابه در نزدیکی یکدیگر، به عنوان یک object در نظر گرفته می شوند نه به صورت مجزا. از آنجا که آمار منطقه شامل هر دو مناطق و محدوده جعبه می شود، اندازه گیری تراکم می تواند فراهم شود. ادغام ابتکاری مناطق بر روی جفت عملیاتی انجام می شود که اگر ادغام شوند دارای تراکم بالاتری نسبت به آستانه در نظر گرفته شده برای هر رنگ به صورت فردی هستند. در مثال بالا، ناحیه ای که دو منطقه را از هم جدا می کند کوچک است، پس تراکم هنگامی که منطقه های ادغام شوند همچنان

می تواند بالا باشد، بنابراین احتمالا تراکم آنها بالاتر از آستانه باشد و ممکن است به عنوان یک منطقه منحصر به فرد گروه بندی شوند.

### ۳ نتایج و کاربردها

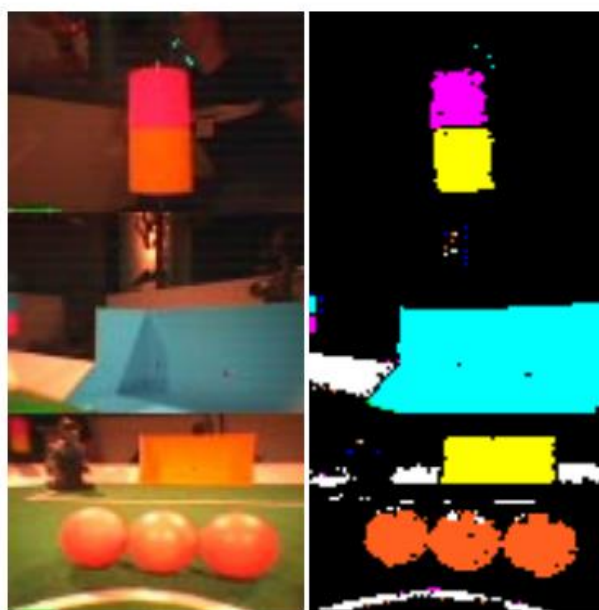
اولین پیاده سازی نمونه ای برای گروهی از رباتهای خودکار ارزان مبتنی بر پلتفرم Probotics Cye بود. این رباتها مبتنی بر قطعات سخت افزاری هستند تا هزینه ها کم باشد و به سادگی کمک کند. آنها همچنان نیازمند عملکرد بینایی بالا هستند زیرا به عنوان سنسور خطر اصلی استفاده می شوند. پلتفرم از یک دوربین NTSC رنگی معمولی متصل به کامپیوتر AMD K6 و یک Winnov دیجیتال استفاده می کند. سیستم عامل مورد استفاده یک توزیع استاندارد لینوکس با درایور برای ضبط ویدئو می باشد. در شکل فعلی آن سیستم می تواند تصاویر  $240 \times 320$  را در ۳۰ هرتز پردازش کند و از ۳۵۰ هرتز CPU استفاده کند.



تصویر ۲: مثالی از چگونگی گروه بندی مناطق پس از run length encoding.

دومین کاربرد موفقیت آمیز برای ورودی Carnegie Mellons به RoboCup-99 لیگ ربات پا دار بود. این رباتها که توسط سونی تولید شده بودند، چهار پا بودند و مشابه نسخه تجاری ربات سرگرمی Aibo بودند. ربات ها فوتبال ۳ به ۳ بازی می کردند. برای اینکه به طور موثری بازی کنند، چندین Object باید شناخته و پردازش شوند، که شامل توپ فوتبال، هم تیمی ها و حریف ها، دو دروازه و ۶ محل علامت گذاری شده اطراف میدان می شد. سخت افزار شامل

یک دوربین بود که تصاویر 88X60 با فضای رنگ YUV در حدود ۱۵ هرتز تولید می کرد. در این کاربرد طبقه بندی رنگ در سخت افزار انجام می گرفت. حتی با اینکه یک مرحله از الگوریتم در نرم افزار کنترل می شود، محدودیت منابع محاسباتی نیازمند الگوریتمی بهینه شده به منظور ماندن زمان برای پردازش های سطح بالاتر مانند کنترل حرکت، رفتار های تیمی و localization است. سیستم مناطق مبتنی بر چگالی ادغام اکتشافی را برای غلبه بر تصاویر noisy با اتصال ساده نمی توانست کنترل کند. سیستم ثابت کرد می تواند در رقابت های RoboCup-99 قدرتمند باشند و تیم ما را قادر ساخت تا رتبه سوم در مسابقات بین المللی را کسب کند.



تصویر ۳: مثالی از یک تصویر طبقه بندی شده با استفاده از رویکرد ارائه شده در مقاله. تصویر سمت چپ ترکیبی از object های ردیابی شده توسط یک ربات فوتبالیست در مسابقات RoboCup-99 است: یک سازنده موقعیت (بالا)، ناحیه دروازه (وسط)، سه توپ فوتبال (پایین). تصویر طبقه بندی شده در سمت راست است.

سومین کاربرد سیستم به عنوان قسمتی از یک ورودی برای لیگ رباتهای کوچک RoboCup است. این دامنه شامل یک دوربین ثابت که از راه دور رباتهایی که بازی میکنند را ردیابی و کنترل میکند می باشد. سیستم ما از یک دوربین در بالای مرکز زمین استفاده می کند. سیستم می تواند توپ و ۱۰ ربات را ردیابی کند. ۵ ربات هم تیم شامل یک نشانگر استاندارد و همچنین یک جهت گیری خاص و تکه رنگ های منحصر به فرد می باشد. ربات های حریف از یک

نشانگر استاندارد و الگوهای اضافی از تصمیم های خودشان برای ردیابی سیستم استفاده می کنند. پردازش Full Frame در این زمینه به دلیل انسداد موقت توپ یا رباتها و ردیابی دقیق توپ و همچنین ردیابی دقیق توپ و رباتهای در حال حرکت با سرعت ۴ متر بر ثانیه مفید است. با رزولوشن 640X480 و تصویر برداری با فرکانس ۳۰ هرتز، سیستم از ۶۰ درصد از Pentium 3 هفت هزار مگا هرتز استفاده می کند.

#### ۴ نتیجه گیری

ما سیستم جدید برای قطعه بندی بلادرنگ برای رنگ تصاویر ارائه دادیم. این سیستم می تواند هر پیکسل در تصویر رنگی را با رزولوشن کامل طبقه بندی کند، مناطق را یافته و تا ۳۲ رنگ طبقه بندی کند، کادر و ناحیه در ۳۰ هرتز را محصور کند. سهم اصلی این سیستم این است رویکردی صرفاً نرم افزاریست که روی سخت افزاری ارزان و همه منظوره پیاده سازی شده است. در میان سیستم های پردازشی Full Frame، این سیستم مزیت قابل توجهی نسبت به راه حل های مبتنی بر صرفاً سخت افزار دارد.

سیستم در چندین مرحله روی تصویر عملیات انجام می دهد:

۱- به صورت اختیاری فضای رنگ را انتخاب می کند.

۲- هر پیکسل را به عنوان یکی از ۳۲ رنگ طبقه بندی می کند.

۳- اجرای length encode هر خط پیمایش بر اساس رنگ.

۴- گروه بندی run های همان رنگ به مناطق

۵- انتقال ساختار جمع آوری شده از آمار مناطق

۶- مرتب کردن مناطق براساس رنگ و اندازه.

سرعت رویکرد ما با توجه به تمرکز بر کارآمد بودن الگوریتم در هر مرحله است. مرحله اول با یک تبدیل خطی انجام می شود. مرحله دوم به جای استفاده از یک رویکرد ساده که ممکن است نیاز به ۱۹۲ مقایسه برای هر پیکسل داشته باشد در عوض از عملیات بیتی AND استفاده می کنیم. مرحله سوم تعداد پیکسل ها خطی است. مرحله چهارم با

استفاده از الحاق یافت شده توسط الگوریتم انجام می شود. مرتب سازی در مرحله پنجم با مرتب سازی مبنایی انجام می شود در حالیکه مرحله ششم با گذر ساده از نتایج داده های ساخت یافته به اتمام می رسد.

این رویکرد به منظور سرعت بخشیدن به دید سطح پایین برای استفاده در نرم افزار های بلادرنگ است که افزایش سرعت سخت افزار یا بسیار گران است یا کارکرد مناسب در دسترس نیست. کارکرد به منظور ارائه با ورودی های با سطح بالاتر که دامنه خاصی را رمز گذاری می کنند مناسب است. این ابزار پردازش های آفلاین قبلی را قادر می سازد تا به عنوان بخشی از یک سیستم بینایی واقعی اجرا شوند. سیستم جاری و انواع آن با موفقیت در سه سخت افزار نشان داده شده اند.

## References

- [1] H. Kitano, Y. Kuniyoshi, I. Noda, M. Asada, H. Matsubara, and E. Osawa. RoboCup: A challenge problem for AI. *AI Magazine*, 18(1), pages 73–85, 1997.
- [2] C. E. Brodley and P. E. Utgoff. Multivariate decision trees. *Machine Learning*, 1995.
- [3] T. A. Brown and J. Koplowitz. The weighted nearest neighbor rule for class dependent sample sizes. *IEEE Transactions on Information Theory*, pages 617–619, 1979.
- [4] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill, 1995.
- [5] Newton Laboratories. Cognachrome image capture device. <http://www.newtonlabs.com>, 1999.
- [6] R. E. Tarjan. Data structures and network algorithms. *Data Structures and Network Algorithms*, 1983.
- [7] The Probotics Cye Personal Robot. <http://www.probotics.com>, 2000.
- [8] M. Veloso, E. Winner, S. Lenser, J. Bruce, and T. Balch. Vision-Servoed Localization and Behaviors for an Autonomous Quadruped Legged Robot *Artificial Intelligence Planning Systems*, 2000.



این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

لیست مقالات ترجمه شده ✓

لیست مقالات ترجمه شده رایگان ✓

لیست جدیدترین مقالات انگلیسی ISI ✓

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی