تفکیک ساختارمند تصاویر راهرو برای هدایت ربات

الهه عمید[۱]، آرش عظیم‌زاده ایرانی[۲] و رضا پورقلی[۳،*]

اطلاعات مقاله

نوع مقاله: پژوهشی
دریافت مقاله: ۱۳۹۹/۱۱/۱۶
پذیرش مقاله: ۱۴۰۰/۱۱/۰۴

واژگان کلیدی:
هدایت ربات،
مولفه‌های ساختارساز راهرو،
تفکیک تصویر.

چکیده

استخراج ساختاری از فضای اطراف ربات به عنوان مسئله‌ای حائز اهمیت برای هدایت ربات در علم پردازش تصویر و بینایی کامپیوتر است که در سال‌های اخیر مورد توجه خاصی قرار گرفته است. برای این منظور بسیار واجب است که ساختار فضای اطراف ربات در محیط مشخص گردد. در روش پیشنهادی از یک دوربین عکاسی برای عکس برداری از راهرو به عنوان فضایی که ربات در آن حرکت می‌کند، استفاده می‌شود. برای استخراج ساختار محیط، در روش پیشنهادی تکنیکی بصورت ترکیبی از نشانه‌های بصری برای ارزیابی احتمال وجود خطوط سطحی لبه‌های عمودی و افقی که به ترتیب متعلق به دیوار و زمین است، استفاده می‌شود. ابتدا از یک نوع الگوریتم تشخیص لبه به نام سوبل برای پیدا کردن خطوط لبه‌دار در دو جهت عمودی و افقی استفاده می‌شود. سپس، یک حد آستانه را برای کاهش خطوط لبه اضافی در نظر می‌گیریم. در نهایت، به منظور قدرتمندسازی روش خود از الگوریتم تفکیک K-means، برای خوشه‌بندی استفاده می‌کنیم. این سه روش در کنار هم یک سیستم قوی را ایجاد می‌کند که کمک می‌کند مؤلفه‌های ساختارساز راهرو از قبیل دیوار، کف، مرز دیوار-کف شناسایی شوند. یافتن این مؤلفه‌ها برای هدایت ربات در محیط‌های داخلی امری بسیار ضروری می‌باشد.

۱-مقدمه

از کاربردهای بینایی کامپیوتر می‌توان به شناسایی و بازشناسایی کاراکترها، تشخیص چهره و هدایت ربات اشاره کرد [۱]. در این مقاله سعی بر آن داریم تا با استفاده از بینایی کامپیوتر به ساختاری از فضا برای هدایت ربات دست یابیم.

ربات برای حرکت، نیازمند درک نسبی از فضای اطراف خود می‌باشد. به همین منزله باید بتواند اطلاعاتی در راستای پیدا کردن موانع و سطحی که در آن به حرکت می‌پردازد، به دست آورد. فضای انتخابی در این پژوهش یک محیط داخلی از نوع راهرو است. در چنین محیطی باید بتوانیم سطح راهرو را مشخص کنیم و دیواره‌ها را تشخیص دهیم. دیواره‌ها شامل هر شی عمودی می‌باشد که معمولاً در اطراف سطح افقی راهرو قرار گرفته است که از جمله آنها می‌توان به در، پنجره، تخته اعلانات و هر شی عمودی اشاره کرد. سطح راهرو نیز با توجه به نوع و جنس، دارای رنگ و بافتی خاص است. نکته حائز اهمیت این است که بتوانیم سطح راهرو را از دیواره‌ها تمیز دهیم و بافت راهرو به اشتباه به عنوان مانع در نظر گرفته نشود. در روش پیشنهادی، تکنیکی بصورت ترکیبی از نشانه‌های بصری برای ارزیابی احتمال وجود خطوط سطحی لبه‌های افقی و عمودی بکار می‌رود به گونه‌ای که ابتدا از الگوریتم تشخیص لبه سوبل برای شناسایی خطوط لبه‌دار استفاده می‌شود. سپس، یک حد آستانه را برای کاهش خطوط اضافی و نگه‌داشتن خطوط لبه کارآمد که نمایانگر دیواره‌ها و زمین است، استفاده می‌کنیم. در نهایت با استفاده از یک الگوریتم

* پست الکترونیک نویسنده مسئول: pourgholi@du.ac.ir
۱. دانش‌آموخته ارشد علوم کامپیوتر، دانشکده ریاضی و علوم کامپیوتر، دانشگاه دامغان
۲. استادیار، دانشکده ریاضی و علوم کامپیوتر، دانشگاه دامغان
۳. استاد، دانشکده ریاضی و علوم کامپیوتر، دانشگاه دامغان

DOI: 10.22075/jme.2022.22576.2038

خوشه‌یابی به بخش‌بندی تصویر برای پیدا کردن مناطق همسان می‌پردازیم و سپس سعی بر آن داریم با ترکیب دو روش تشخیص لبه و بخش‌بندی تصویر به بهترین عملکرد خروجی برای یافتن مولفه‌های دیوار، کف و مرز دیوار-کف برسیم.

۱-۱-بیان مسئله

اولین سوالی که به ذهن شخصی که قصد ارتقاء یا بهبود یک تکنیک خاص را دارد، ممکن است برسد احتمالاً کاستی های یک تکنیک است. علاوه بر آن، تنها با دانستن مضرات و ضعف‌های یک تکنیک می‌توان راه‌حل مناسبی را برای حل آن پیشنهاد کرد. بنابراین برای حرکت ربات در محیط داخلی در پایه‌ای ترین مرحله باید بتوانیم محیط اطراف ربات را درک کنیم و این مقوله جز با دانستن ساختار راهرو امکان پذیر نیست. برای این منظور از روش‌های تشخیص لبه و بخش‌بندی استفاده می‌شود.

انواع روش‌های تشخیص لبه به استخراج اکثر خطوط لبه در تصویر می‌پردازند [۲ و ۳]. بنابراین مرزهای زمین-دیوار به تنهایی بوسیله این روش نمی‌توانند پیدا شوند. مسئله‌ای که در پی حل آن برمی‌آییم این است که چگونه می‌توانیم خطوط لبه‌های مربوط به دیوار و سطح راهرو را از بین تمام خطوط لبه به دست آمده، انتخاب کنیم.

برای تشخیص لبه‌ها از الگوریتم سوبل استفاده شده است، این الگوریتم به پیدا کردن لبه‌ها در راستای افقی و عمودی می‌پردازد [۴]، لبه‌های عمودی و افقی به ترتیب نمایانگر دیوار و کف هستند. بعد از عملیات تشخیص لبه، برای صحت تعلق خطوط لبه پیدا شده به اشیا در کف و بر روی دیوار و حذف شلوغی بیش از حد لبه‌ها، ابتدا این خطوط لبه، باید از یک حد آستانه به منظور استخراج خطوط لبه مرتبط گذرکنند، خطوط لبه غیرمرتبط به وسیله یکی از روش‌های آستانه‌گذاری معرفی‌شده در روش پیشنهادی حذف می‌شوند و اینگونه از تاثیر داده‌های غیرضروری جلوگیری کرده و همچنین تعداد محاسبات را کاهش می دهیم.

تفکیک به فرآیند تقسیم‌بندی یک تصویر به چندین منطقه از مجموعه پیکسل‌ها گفته می‌شود که هر منطقه با سایر مناطق از نظر خصوصیاتی مانند رنگ، شدت، بافت، فضا و ... تفاوت دارد، در واقع تفکیک بدین منظور صورت می‌گیرد که یک تصویر از پیچیدگی‌های ظاهری به تصویری ساده‌تر و قابل فهم‌تر تبدیل شود که در نتیجه تجزیه و تحلیل آن تصویر آسان‌تر می‌گردد. وظیفه الگوریتم تفکیک (بر پایه خوشه‌بندی) یافتن مناطق همگن است، هرکدام از این مناطق می‌توانند یک شیء در تصویر باشند [۵].

۱-۲-پیشینه تحقیق

برای تمایز و تشخیص بین اشیاء موجود در یک تصویر، می توان از الگوریتم تقسیم‌بندی بر اساس سه روش تشخیص لبه، آستانه گذاری و الگوریتم خوشه‌بندی استفاده کرد [۳]. به منظور تعیین ساختار راهرو برای هدایت ربات نیازمند به دست آوردن اطلاعات مفید از تصاویر هستیم. در همین راستا از روش تشخیص لبه به پیدا کردن لبه‌ها می‌پردازیم [۶].

تصویر تابعی دو بعدی است و لبه زمانی بوجود می‌آید که تغییرات در شدت روشنایی داشته باشیم. در نتیجه می توانیم با استفاده از مشتق‌گیری، مکان‌هایی را که پتانسیل لبه بودن دارند یا می‌توانند داشته باشند پیدا کنیم [۲].

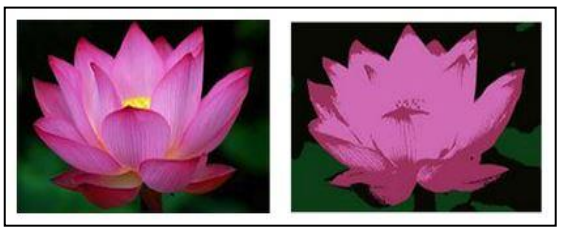
در الگوریتم‌های تشخیص لبه، برای پیدا کردن لبه‌ها از عملگرهای ویژه‌ای استفاده می‌شود، یکی از این عملگرها کنی است [۷]. جان کنی روش شناسایی اشیاء موجود در تصویر را پیشنهاد کرده است که این روش ابتدا تصویر را به یک تصویر خاکستری تبدیل می‌کند و در مراحل بعدی، خطوط لبه را پیدا می‌کند و سرانجام با ایجاد آستانه‌گذاری آن را به پایان می‌رساند. این الگوریتم دارای ۳ بخش اصلی است: ۱-تضعیف نویز، ۲-پیدا کردن نقاطی که بتوان آن‌ها را به عنوان لبه در نظر گرفت، ۳- حذف نقاطی که احتمال لبه بودنشان کم است [۷ و ۸]. همچنین از عملگر پری‌ویت لبه‌های افقی و لبه‌های عمودی را پیدا می‌کند. پری‌ویت از دو ماسک برای تشخیص لبه در جهت افقی و دیگری برای تشخیص لبه در جهت عمودی استفاده می‌کند [۹].

عملگر سوبل هم مانند پری‌ویت عمل می‌کند با این تفاوت که ضرایب ماسک‌های افقی و عمودی سوبل ثابت نیستند و قابل تنظیم می‌باشند. همچنین، سوبل در حذف نویز که مسئله‌ای مهم در مشتق گرفتن است، بهتر عمل می‌کند. از دیگر تفاوت‌های سوبل و پری‌ویت این است که عملگر پری ویت بر روی تصاویر رنگی عمل نمی‌کند [۱۰].

پس از محاسبه مشتق، می‌توانیم آستانه‌ای را برای کاهش تعداد لبه‌ها تعریف کنیم و لبه‌های موثرتر را با توجه به نیاز و کاربرد نگه داریم [۱۱].

از الگوریتم‌های تفکیک تصویر می‌توان به الگوریتم K-means اشاره کرد که خوشه‌بندی مناسبی را ایجاد می

کند [۱۲]. این الگوریتم شروع به خوشه‌بندی با تعیین تعداد k نقطه مرکزی اولیه، به صورت تصادفی می‌کند. سپس هر پیکسل تصویر را در تحت نزدیک‌ترین نقطه مرکزی آن قرار می‌دهد. پس از آن، نقاط مرکزی جدید را با میانگین‌گیری پیکسل‌های تحت هر نقطه مرکزی محاسبه می‌کند و این دو مرحله الگوریتم پیشین را به طور متناوب تا همگرایی تکرار می‌کند [۱۳]. شکل (۱) نمایشی از خوشه بندی تصویر توسط روش k-means است.



شکل ۱. خروجی روش K-means ، تصویر سمت چپ عکس اصلی و تصویر سمت راست عکس خوشه‌بندی شده با روش K-means و K=4 است.

همانند سایر روش‌های خوشه بندی، روش K-means هم ضعف‌هایی را درون خود دارد که از آن‌ها می‌توان به زمان محاسبه بالا، تعیین تعداد خوشه‌ها توسط کاربر و وجود داده‌های پرت و در نتیجه پایین آمدن عملکرد خوشه‌بندی اشاره کرد [۱۴]. برای جبران یکی از ضعف‌های روش K-means که زمان محاسبه بالاست، از یک ساختار برای نگهداری فاصله نقاط تا نقاط مرکزی استفاده می‌شود. روال کار به این صورت است که برای هر نقطه می‌توان فاصله نقطه را با نزدیک‌ترین خوشه موردنظر نگه داشت و در یک ساختار ذخیره کرد. سپس، برای تکرار بعدی، ابتدا فاصله نقطه تا نزدیک‌ترین خوشه قبلی محاسبه می‌شود. اگر فاصله کمتر یا مساوی فاصله قبلی بود، نقطه در خوشه قبلی باقی می‌ماند و دیگر نیازی به محاسبه فاصله بین این نقطه و تمام خوشه‌های دیگر نیست. در غیر این‌صورت فاصله داده با دیگر نقاط مرکزی محاسبه می‌شود وکوچک‌ترین فاصله انتخاب می‌شود و جایگزین مقدار قبلی می‌شود. این گونه زمان محاسبه کاهش می‌یابد. این ایده از آنجا سرچشمه می گیرد که الگوریتم K-means خوشه‌های کروی را کشف می‌کند و نقاط مرکزی موجود در داده‌ها علاقه دارند که به مرکز ثقل خود که همان مرکز است نزدیک شوند. در نتیجه نقاط و مراکز هر خوشه به هم نزدیک می‌شوند. بنابراین، نیازی به محاسبه فاصله تا سایر مراکز خوشه دیگر نیست [۱۵].

از دیگر الگوریتم‌های کارشده در رابطه با تفکیک و خوشه‌بندی می‌توان به الگوریتم MST-Based Segmentation اشاره‌کرد که تلاش می‌کند با بخش‌بندی یک عکس به تعدادی منطقه سعی در ساده‌سازی عکس و نشان دادن اشیا با درکی آسان‌تر بپردازد. MST-Based Segmentation این‌گونه کار می‌کند که ابتدا با استفاده از نمایش مبتنی بر گراف تصویر، یک پیش‌فرض دال بر وجود مرز بین دو منطقه تعریف می‌شود. سپس، از یک گریدبندی تصویر برای تعریف همسایه‌های محلی بین پیکسل‌های تصویر استفاده می‌کند و تفاوت در شدت یا رنگ را بین هر دو جفت همسایه اندازه می‌گیرد و در ادامه پیکسل‌های تصویر را به نقاط در فضای ویژگی مپ می‌کند که مختصات (x,y) و مقدار (r,g,b) را ترکیب می‌کند. لبه‌های موجود در گراف، نقاطی را به هم وصل می‌کند که در فضای ویژگی به هم نزدیک باشند. سپس با یک الگوریتم کارآمد تفکیک را براساس همین پیش‌فرض توسعه می‌دهد و بوسیله همین روش، جزئیات را در مناطقی که متغیرهای کمی دارند حفظ می‌کند و در مناطقی که متغیرهای زیادی دارند نادیده می گیرد [۱۶].

در روش K-means معمول، ممکن است هنگام پایان یافتن خوشه‌بندی، تصویر بیش از حد بخش‌بندی شود. همچنین، ممکن است پیکسل یا منطقه‌ای در بخش‌بندی تأثیر به سزایی نگذارند [۱۷]. برای حل این مشکل می‌توان با استفاده از الگوریتم Spatial Constrained K-means تفکیک مناطق انجام شود که خاصیت تناوبی این الگوریتم منجر به گسترش پایدار مناطق می‌شود. در نهایت در کنار خوشه‌های با معنی، خوشه‌های کوچک بی‌معنی نیز بوجود می‌آیند که این مناطق باید تصفیه شوند. همچنین، باید مناطق بسیار نازک و کوچک نیز بر اساس شباهت رنگی و نسبت لبه به مناطق بزرگ‌تر بپیوندند [۱۸].

بعضی از رویکردهای پیشین، برای هدایت ربات به بررسی مکان ربات می‌پردازند. به گونه‌ای که ربات در مرکز راهرو بماند، یکی از این پارامترها محاسبه فاصله تا انتهای راهرو است و دیگری برآورد جهت راهرو می‌باشد [۴].

رویکرد دیگر برای پیدا کردن انتهای راهرو از سه مرحله تشکیل شده است. مرحله اول خطوط عمودی راهرو را تماماً استخراج می‌کند. سپس، در مرحله میانی چندین بخش

خط که بوسیله مرحله قبل به‌دست‌آمده است را با الگوریتم Subtractive Clustering ترکیب می‌کند. در مرحله پایانی، خطوط عمود استخراج شده از مرحله اول را توسط شماری از خطوط عمودی که نقطه انتهایی آن بر روی خطوط پیش‌بینی شده راهرو قرار دارند، تائید یا رد می‌کند. در انتها، نقطه محوشونده توسط Subtractive Clustering تخمین زده می‌شود و سپس راهرو را به دو گروه سمت راست و چپ تقسیم می‌کند [۱۹].

۱-۳-اهداف مسئله

هدف اصلی در این روش به دست آوردن مؤلفه‌های ساختارساز راهرو است. برای دستیابی به این هدف باید اجزای راهرو را پیدا کنیم. در نتیجه ما از ابتدا هدف خود را به سه بخش تقسیم می‌کنیم: ۱- تشخیص لبه، ۲- تفکیک و بخش‌بندی تصویر ۳- تلفیق مناسبی از مرحله ۱ و ۲ به منظور به دست آوردن مرزبندی متناسبی از مؤلفه‌های عمودی و افقی یا جدایی مناسبی از اجزای تصویر.

تشخیص لبه از تکنیک‌های پردازش تصویر است که در بینایی کامپیوتر بکار می‌رود. لبه‌ها در تصویر تغییرات در شدت روشنایی و یا رنگ هستند که نمایانگر تغییری در تصویر مانند شی و پس‌زمینه می‌باشند. برای یافتن مرز دیوار و زمین از تشخیص لبه استفاده می‌شود به طوری که خطوط پیداشده در راستای افق به زمین و خطوط عمودی به دیوار مربوط هستند. در این رویکرد که برای تشخیص لبه‌ها از الگوریتم سوبل استفاده شده است به پیدا کردن لبه ها در راستای افقی و عمودی پرداخته می‌شود. همانطور که گفته شد، لبه‌های عمودی و افقی به ترتیب نمایانگر دیوار و کف هستند. چالش دیگر پیشرو ازدیاد خطوط لبه و عدم نیاز به همه لبه‌های پیدا شده است. پس در درجه اول باید خطوط لبه بی‌فایده را حذف کنیم. در نتیجه به‌وسیله یکی از روش‌های آستانه‌گذاری معرفی شده، خطوط غیرمرتبط که توانایی نمایش خطوط در راستای افق و عمود را ندارند، ازبین می‌بریم و اینگونه از تأثیر داده‌های غیرضروری جلوگیری کرده و تعداد محاسبات را کاهش می‌دهیم. همچنین برای مناطقی که نمی‌توان به‌تنهایی با کمک لبه های افقی و عمودی به مرز دیوار و کف پی برد، از تلاقی خطوط عمودی و افقی کمک می‌گیریم و مرزبندی از کف و دیوار را می‌سازیم.

تفکیک از چالش‌های کلیدی در جهت استخراج مفاهیم از تصاویر است که از روش‌های پردازش تصویر و بینایی کامپیوتر بصورت قابل توجهی در این حوزه استفاده می‌کند و تلاش می‌کند به بازیابی محتوا بر پایه تصویر بپردازد. از روش‌های تفکیک و بخش‌بندی تصویر می‌توان به خوشه بندی بر پایه K-means اشاره کرد. در این رویکرد با توجه به تمایل برای بهبود عملکرد روش K-means به توسعه ویژگی‌های این روش می‌پردازیم و با درک محدودیت‌های روش مورد نظر همچون هزینه بالا محاسبات و برای جلوگیری از ایجاد خوشه‌های خیلی بزرگ و یا خیلی کوچک به اضافه‌کردن مؤلفه‌های جدیدی به روش K-means می‌پردازیم. اول خوشه‌بندی دقیق‌تری را با در نظر گرفتن بافت به عنوان ویژگی دیگری علاوه بر ویژگی رنگ، ارائه می‌دهیم. دوم از شرایط توقف (تعداد تکرار، اندازه کلاستر) برای متوقف کردن فرآیند خوشه‌بندی استفاده می کنیم. هدف این است که از تشکیل خوشه‌های خیلی بزرگ یا خیلی کوچک جلوگیری شود.

در نهایت به ترکیب این دو روش اصلی می‌پردازیم تا با پوشاندن کمی و کاستی‌های این دو روش بتوان تفکیک ساختارمندی از راهرو برای هدایت ربات را پیدا کرد.

۲-راه حل پیشنهادی

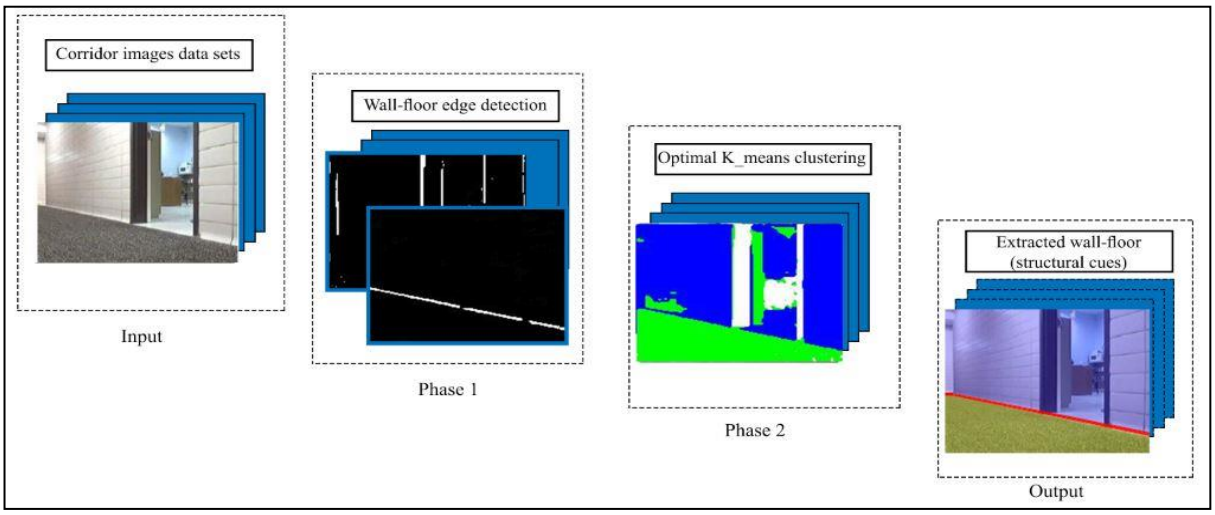
در این مقاله، یک رویکرد مبتنی بر پردازش تصویر و بینایی کامپیوتر به منظور شناسایی مؤلفه‌های راهرو پیشنهاد شده است. در شکل (۲)، نمای کلی از روش‌های پیاده‌سازی و الگوریتم‌های مورد استفاده در رویکرد پیشنهادی معماری بخش‌بندی این مقاله را مشاهده می‌فرمائید که در قسمت‌های آینده به تفصیل به معرفی و توضیح آن‌ها خواهیم پرداخت.

نوع ورودی ما در این مقاله یک مجموعه‌ای از داده‌ها از نوع تصویر و مربوط به محیط داخلی ساختمان از نوع راهرو است.

مجموعه داده‌های تصاویر ورودی شامل مجموعه داده‌هایی از تصاویر راهروهای دانشگاه دامغان و مجموعه داده‌هایی از تصاویر تز Yinxiao Li است [۳].

روش کار به این صورت است که ابتدا دیتاستی از تصاویر راهرو به عنوان ورودی توسط یک دوربین متصل به ربات در محیط راهرو گرفته می‌شود و سپس به منظور پردازش و اعمال روش‌های لبه‌یابی، خوشه‌بندی و ... به الگوریتم‌های متناظر به عنوان ورودی داده می‌شود.

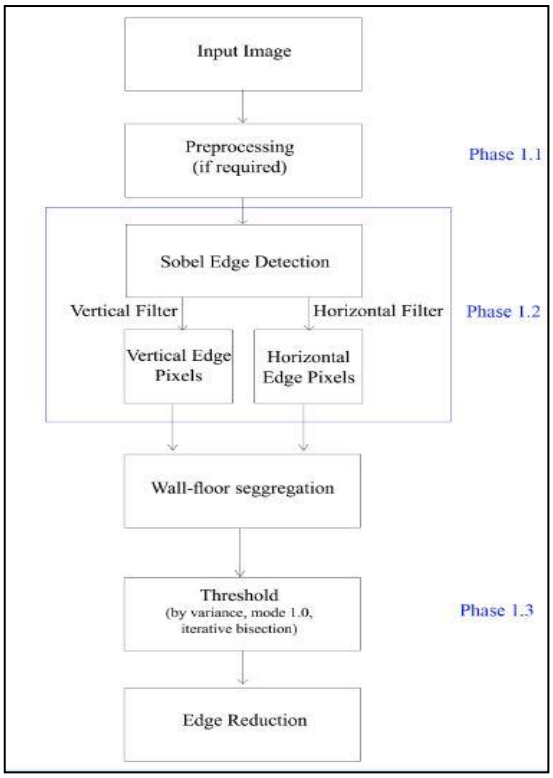
نکته حائز اهمیت که در روش پیشنهادی، استفاده از تصاویر با وضوح پایین است. چرا که عملکرد سیستم بصری چشم و مغز انسان ثابت کرده‌است که مغز انسان نیازمند تصاویر با وضوح بالا برای دیدن و سپس درک اجسام در محیط نیست، پس با الهام از این قضیه و همچنین زمان‌بر بودن اجرای الگوریتم‌ها بر روی تصاویر با کیفیت بالا، کیفیت تصاویر را به میزان لازم تا زمانی که ساختار اساسی تشکیل دهنده و عناصر اصلی تصویر تغییر خاص یا جابجایی چشمگیری نداشته باشد پایین می‌آوریم، بدین وسیله هم سرعت اجرای الگوریتم را بالا برده‌ایم و هم در زمان محاسبه صرفه جویی کردیم.



شکل ۲. نمای کلی از معماری بخش‌بندی

۲-۱- نمای کلی تشخیص لبه در راستای دیوار-زمین

نمای کلی معماری تشخیص لبه را در فلوچارت طراحی شده در شکل (۳) مشاهده می‌فرمائید.



شکل ۳. نمای کلی تشخیص لبه در راستای دیوار-زمین

فلوچارت پیشنهادی از ۳ بخش اصلی شامل پیش پردازش، تشخیص لبه بوسیله روش سوبل و آستانه‌گذاری تشکیل شده است. اعداد نام برده شده در فازها بیانگر ارتباط بین این فازها و فازهای نام برده شده در شکل (۲) هستند. بر طبق فلوچارت شکل (۳)، در اولین مرحله از الگوریتم کاهش لبه، الگوریتم ورودی از جنس تصویر دریافت می‌کند، این تصویر می‌تواند رنگی (RGB) و یا سطح خاکستری (GrayScale) باشد.

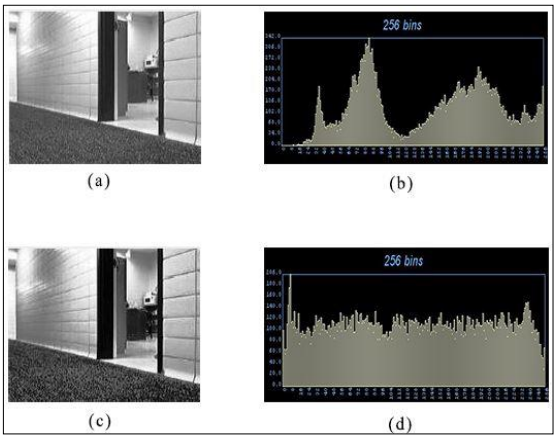
فاز ۱,۱

این فاز اختیاری است و زمانی استفاده می‌شود که تصویر دارای کیفیت پایینی باشد یا زمانی که در تصویر نویز وجود داشته باشد.

وقتی کنتراست تصویر پایین است، به این معناست که تغییرات بین کوچک‌ترین و بزرگ‌ترین شدت روشنایی در تصویر کم و در نتیجه وضوح تصویر پایین است. با تنظیم هیستوگرام تصویر می‌توان کنتراست تصویر را تا جای ممکن بالا برد [۲۰ و ۲۱].

هیستوگرام یک نمودار برای کار با تصاویر دیجیتال است. محور عمودی نشان‌دهنده تعداد پیکسل‌هاست و محور افقی نشان‌دهنده محدوده مقادیر شدت رنگ‌هاست. سمت چپ محور افقی نشان‌دهنده محدوده مشکی و سمت راست آن محدوه سفید و بین این دو، نشان دهنده پیکسل‌های طیف

خاکستری است. تکنیک Histogram Equalization هم برای بالا بردن و هم برای کاهش کنتراست تصویر به ترتیب در نقاط با کنتراست پایین و بالا بکار می‌رود. زمانی که کنتراست تصویر بالا می‌رود، شدت رنگ تصویر نیز افزایش می‌یابد. در نتیجه لبه‌ها و خطوط مرز بهتر مشخص می شوند و جزئیات تصویر بیشتر نمایان می‌شود.
نمونه‌ای از تاثیر Histogram Equalization را در شکل (۴) مشاهده می‌فرمائید.



شکل ۴. Histogram Equalization ، دو شکل (a) و (b) نشان دهنده تصویر و نمودار هیستوگرام آن هستند. شکل (c) و (d) نشان دهنده همان تصویر بعد از اعمال Histogram Equalization بر روی آن است. مشاهده می‌شود که کنتراست تصویر بعد از اعمال روش Histogram Equalization بالاتر رفته است.

فاز ۲,۱

لبه‌ها مجموعه‌ای از پیکسل‌های به هم پیوسته هستند که در آن بصورت ناگهانی و یا آرام تغییرات شدت روشنایی رخ می‌دهد [۷]. تشخیص لبه یک روش پردازش تصویر است که به پیدا کردن مرز اشیا در تصویر می‌پردازد و بر پایه عدم پیوستگی در تغییرات شدت روشنایی استوار است. از تشخیص لبه برای تشخیص تغییرات شدید در روشنایی، تشخیص اشیا و بخش‌بندی عکس استفاده می‌شود.
لبه‌ها نشان‌دهنده مرز بین اشیا و پس‌زمینه و به بیانی محل وجود تغییرات در تصویر هستند. تغییرات در ریاضیات بوسیله مشتق نشان داده می‌شود، از آنجایی که تصویر تابعی دوبعدی و متشکل از پیکسل‌هایی گسسته است، می‌توان با مشتق‌گیری جزئی، مکان‌هایی که تابع تغییر ناگهانی دارد را مشخص کرد، این مکان‌ها، لبه هستند. برای به دست آوردن مشتق باید از تابع تصویر یک‌بار در راستای x و یک‌بار در راستای y مشتق تک بعدی مرتبه اول گرفت، در ادامه به ترتیب معادله ۱ مشتق تابع f ($\frac{\partial f}{\partial x}$) در راستای x و معادله ۲ مشتق تابع f ($\frac{\partial f}{\partial y}$) در راستای y است:

$$\frac{\partial f}{\partial x} = f\,(x+1) - f\,(x) \qquad (۱)$$

$$\frac{\partial f}{\partial y} = f\,(y+1) - f\,(y) \qquad (۲)$$

دیوار و هر شیء که به دیوار مربوط است بر روی محور عمودی تصویر قرار می‌گیرد و بلاطبع زمین و کف نیز بر روی محور افقی قرار دارند. اگر بتوانیم خطوط لبه افقی و عمودی را در تصویر پیدا کنیم، می‌توانیم به محدوده دیوار-زمین دست یابیم. در این مقاله از فیلتر سوبل که از انواع عملگرهای تشخیص لبه است، برای پیداکردن پیکسل‌های لبه توسط مشتق مرتبه اول استفاده می‌کنیم. عملگر سوبل از دو نوع ماسک مشتق برای لبه‌یابی استفاده می‌کند.
سوبل یک فیلتر مکانی است و لبه‌ها در تصویر را با استفاده از ماسک فیلترها در دو جهت افقی و عمودی استخراج می کند. ماسک‌های سوبل بر روی پیکسل‌های تصویر حرکت می‌کنند و با عبور از هر پیکسل تصویر، مجموع ضرب ماتریس ماسک و ماتریس تصویر برای هر پیکسل محاسبه می‌شود. به این عملیات کانوولوشن گفته می‌شود. سپس خروجی لبه‌ها را در تصویر جدیدی نمایش می‌دهند.
مشتق گیری در راستای افق G_x و در راستای عمود G_y با استفاده از فیلترهای سوبل با روش spatial filtering انجام شده است.گرادیانت (تغییرات درجه روشنایی که بوسیله مشتق نوع یک قابل اندازه گیری است) در راستای افق G_x خطوط لبه افقی و گرادیانت در راستای عمود G_y خطوط لبه عمودی را نشان میدهد. نحوه کانوولوشن کردن فیلتر سوبل بر روی پیکسل‌های تصویر که در نهایت منجر به مشتق‌گیری از تصویر می‌شود که به صورت معادله ۳ نمایش داده می‌شود:

$$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f\,(x+s, y+t) \qquad (۳)$$

در این معادله w ماسک و f تصویر است، همواره پیکسل مرکزی ماسک از روی تمام پیکسل‌های تصویر دیجیتال عبور کرده و مقادیر جدید را برای آن‌ها محاسبه و سپس جایگزین می‌کند.

در تصویر خروجی لبه‌یابی شده توسط فیلتر سوبل، خطوط عمودی پتانسیل دیوار بودن و خطوط افقی پتانسیل زمین یا کف بودن را دارند.

معادله ۴ و ۵ به‌ترتیب عملیات کانوولوشن ماتریس ماسک سوبل بر روی تصویر و محاسبه مشتق (G_x) در راستای x و مشتق (G_y) در راستای y را نشان می‌دهد:

$$G_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \times A \qquad (۴)$$

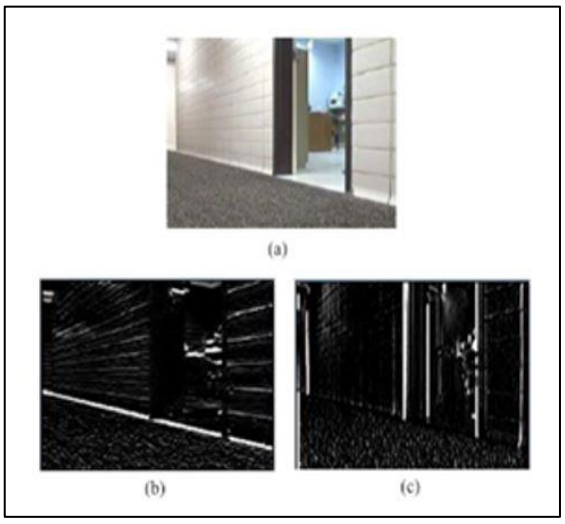
$$G_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \times A \qquad (۵)$$

همچنین معادله ۶ و ۷ برای محاسبه اندازه گرادیان (Gradient magnitude) و جهت گرادیان بکار می‌روند.

$$G = \sqrt[2]{G_x^{\,2} + G_y^{\,2}} \qquad (۶)$$

$$\theta = \operatorname{arctg} \frac{G_x}{G_y} \qquad (۷)$$

G اندازه گرادیان و θ جهت گرادیان است. در عمل به منظور کاهش هزینه محاسبات می‌توان از تقریب $|G_x| + |G_y|$ بجای $\sqrt[2]{G_x^{\,2} + G_y^{\,2}}$ استفاده کرد. شکل (۵) نشان دهنده اعمال فیلتر سوبل بر روی تصویر است.



شکل ۵، جدایی دیوار کف بر اساس الگوریتم سوبل، (a) تصویر اصلی، (b) پیدا کردن خطوط لبه افقی به وسیله الگوریتم سوبل، (c) پیدا کردن خطوط لبه عمودی به وسیله الگوریتم سوبل

فاز ۳,۱

در فاز ۲,۱ به پیدا کردن خطوط لبه پرداختیم. مشاهده می شود که تمام خطوط لبه به دست آمده، لازم نیستند و بسیاری از آن‌ها به جزئیات بی‌فایده‌ای مانند بازتاب‌های شدید و یا سایه‌ها مربوط می‌شوند، اگر بتوان تعداد لبه‌ها را کاهش داد و لبه‌های مفید را نگه داشت، به تصویر لبه‌یابی شده با لبه‌های کارآمدتری خواهیم رسید. منظور از لبه‌های کارآمدتر، تصویری است که در آن شلوغی بیش از حدی از لبه‌ها نداریم و تنها لبه‌های مهم و مرتبط با دیوار و کف باقی مانده‌اند. بعد از لبه‌یابی، تصویر به دو گروه از پیکسل‌های روشن و تیره تفکیک می‌شود. یک راه برای استخراج اشیا از پس‌زمینه استفاده از آستانه‌گذاری است. در واقع آستانه‌گذاری بر روی یک تصویر سطح خاکستری یا باینری انجام می‌پذیرد و پیکسل‌های مربوط به اشیا و پس زمینه را در دو گروه پیکسل‌های روشن و تیره بوسیله یک حد آستانه T دسته‌بندی می‌کند. سپس، برای هر پیکسل (x,y) اگر f(x,y)>T باشد، آن پیکسل به عنوان شی در نظر گرفته می‌شود. در غیر این صورت، پیکسل مورد نظر از پیکسل‌های پس‌زمینه محسوب می‌شود و در نتیجه اشیا با پیکسل‌هایی با رنگ روشن در پس‌زمینه با پیکسل‌های با رنگ تیره قرار می‌گیرند. تابع T بصورت معادله ۸ محاسبه می‌شود:

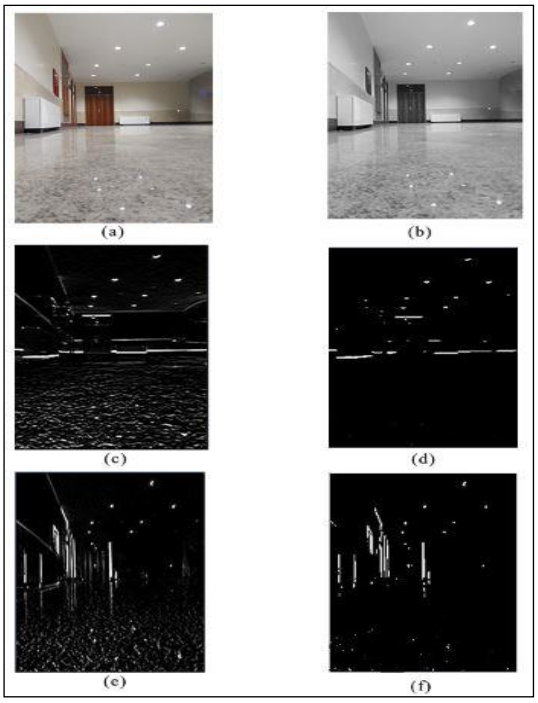
$$T = T[x, y, p(x, y), f(x, y)] \qquad (۸)$$

در آن f(x,y) مقدار سطح خاکستری نقطه (x,y) است و p(x,y) نشان‌دهنده برخی ویژگی‌های محلی، مانند میانگین مقدار سطح خاکستری همسایگی‌ها به مرکزیت پیکسل (x,y) در نظر گرفته می‌شود. تصویر g(x,y) با اعمال آستانه‌گذاری به صورت معادله ۹ محاسبه می‌شود:

$$g(x, y) = \begin{cases} 1 & if \quad f(x, y) > T \\ 0 & if \quad f(x, y) \le T \end{cases} \qquad (۹)$$

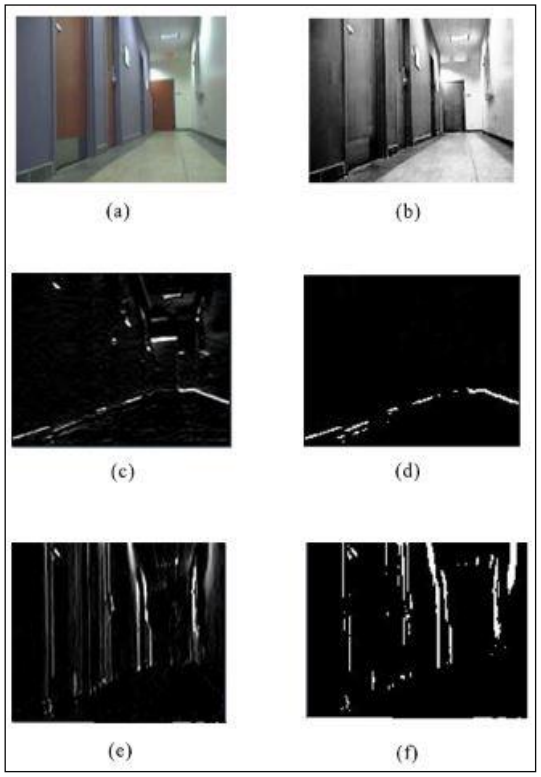
پیکسل‌ها با برچسب ۱ مربوط به اشیا در تصویر هستند و پیکسل‌ها با برچسب ۰ مرتبط با نقاط پس‌زمینه تصویر هستند. در نتیجه بعد لبه‌یابی و به‌دست آوردن خطوط لبه یعنی بعد از جدایی دیوار و زمین بهتر است با تعریف یک حد آستانه از تعداد خطوط لبه کاست و فقط نقاط مفید از لبه‌ها که نشان‌گر اشیا هستند را نگه داشته و سایر نقاط لبه را به عنوان پس‌زمینه تصویر قرار دهیم. در عمل، با استفاده

از آستانه‌گذاری، پیکسل‌هایی که مقادیری بالاتر از مقدار آستانه دارند، به عنوان لبه‌های مفید، برجای می‌مانند و مقادیر پایین‌تر از مقدار آستانه به پس‌زمینه می‌پیوندند، که در نتیجه منجر به کاهش لبه‌ها در تصویر دیجیتال می‌شود. آستانه‌گذاری زمانی خوب عمل می‌کند که رنگ دیوار و زمین متفاوت باشد، مثلاً اگر دیوارها روشن و زمین تیره باشد یا قسمت پایین دیوارها تیره‌تر از خود دیوار باشد، در این صورت مرز دیوار و زمین بهتر مشخص خواهد شد. این تفاوت رنگ‌ها می‌تواند به دلیل وجود قرنیز در قسمت پایین دیوار و یا آلودگی‌های جمع شده در پایین دیوار و کنج‌ها باشد. در رویکرد پیشنهادی، ۳ روش آستانه‌گذاری بکار رفته است: آستانه‌گذاری بر اساس واریانس، آستانه‌گذاری mode 1.0 و آستانه‌گذاری با روش Iterative bisection. کاربر می‌تواند هرکدام از این روش‌ها را به دلخواه خود انتخاب کند و یا به صورت دستی مقدار آستانه را تغییر دهد و نتیجه را به صورت تعاملی مشاهده کند. در نهایت خروجی رویکرد تعریف شده، تصویری لبه‌یابی شده خواهد بود که لبه‌های ناکارای آن کاهش یافته‌اند.



شکل ۶. نمونه خروجی تصویر بعد از اعمال تشخیص لبه شماره ۱ a) تصویر اصلی از راهرو دانشگاه دامغان، b) همان تصویر در سطح خاکستری، c) تصویر لبه‌یابی شده در راستای افقی، d) همان تصویر بعد از آستانه‌گذاری در راستای افقی، e) تصویر لبه‌یابی شده در راستای عمودی، f) همان تصویر بعد از آستانه‌گذاری در راستای عمودی

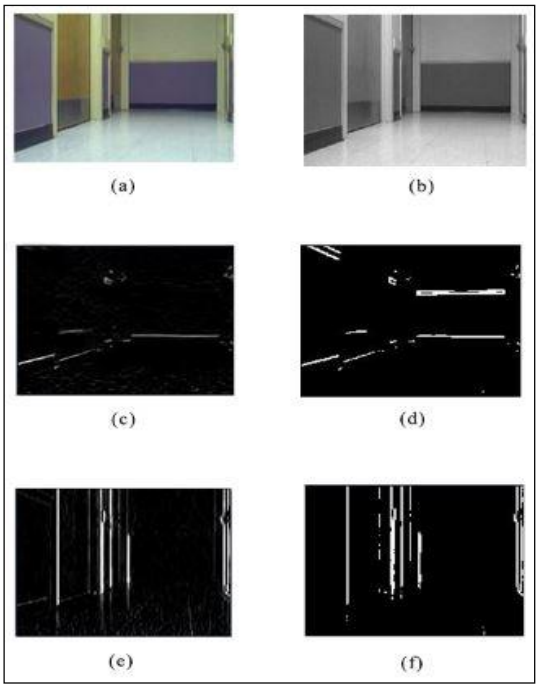
در شکل‌های (۶)، (۷)، (۸)، (۹) و(۱۰) نمونه خروجی‌ها بعد از اعمال روش تشخیص لبه وسپس تحت اعمال کاهش خطوط لبه توسط آستانه‌گذاری مناسب قرار داده شده‌اند و مشاهده می‌شود که از شلوغی بیش از حد لبه‌ها در تصویر کاسته شده است و لبه های مرتبط با دیوار و زمین باقی مانده است.



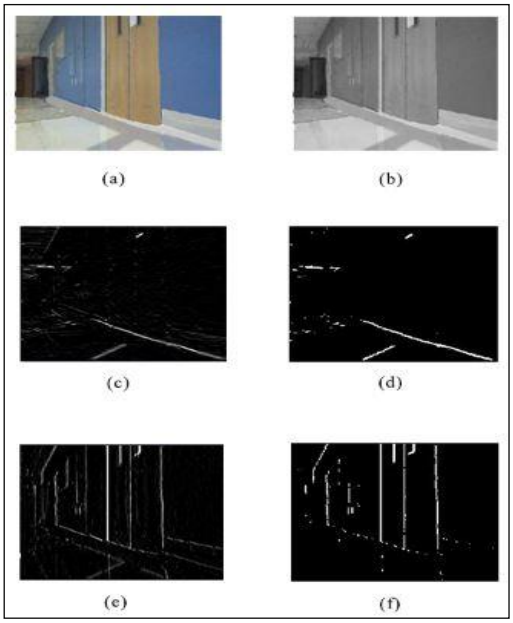
شکل ۷، نمونه خروجی تصویر بعد از اعمال تشخیص لبه شماره ۲، a) تصویر اصلی از راهرو، b) همان تصویر در حالت سطح خاکستری، c) تصویر لبه‌یابی شده در راستای افقی، d) همان تصویر بعد از آستانه‌گذاری در راستای افقی، e) تصویر لبه‌یابی شده در راستای عمودی، f) همان تصویر بعد از آستانه‌گذاری در راستای عمودی

۲-۲-روش خوشه‌بندی بر پایه K-means بهینه

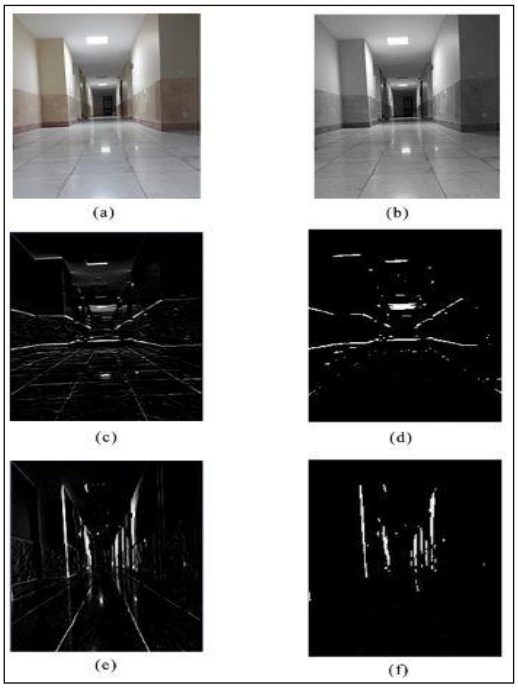
روش K-means یکی از روش‌های خوشه‌بندی در علم داده‌کاوی است. روش K-means را می‌توان توسط الگوریتم‌های مختلفی بیان کرد ولی همه آن‌ها دارای روال تکراری هستند و برای تعداد k خوشه سعی در بدست آوردن نقاطی به عنوان مرکز خوشه‌ها دارند، که این نقاط در واقع همان میانگین نقاط متعلق به هر خوشه هستند. همچنین، در ادامه هر نمونه داده را به یک خوشه نسبت می‌دهند. بطوریکه که آن داده کمترین فاصله را تا مرکز خوشه داشته باشند.



شکل ۸، نمونه خروجی تصویر بعد از اعمال تشخیص لبه شماره ۳، a) تصویر اصلی از راهرو، b) همان تصویر در حالت سطح خاکستری، c) تصویر لبه‌یابی شده در راستای افقی، d) همان تصویر بعد از آستانه‌گذاری در راستای افقی، e) تصویر لبه‌یابی شده در راستای عمودی، f) همان تصویر بعد از آستانه‌گذاری در راستای عمودی



شکل ۹، نمونه خروجی تصویر بعد از اعمال تشخیص لبه شماره ۴، a) تصویر اصلی از راهرو، b) همان تصویر در حالت سطح خاکستری، c) تصویر لبه‌یابی شده در راستای افقی، d) همان تصویر بعد از آستانه‌گذاری در راستای افقی، e) تصویر لبه‌یابی شده در راستای عمودی، f) همان تصویر بعد از آستانه‌گذاری در راستای عمودی



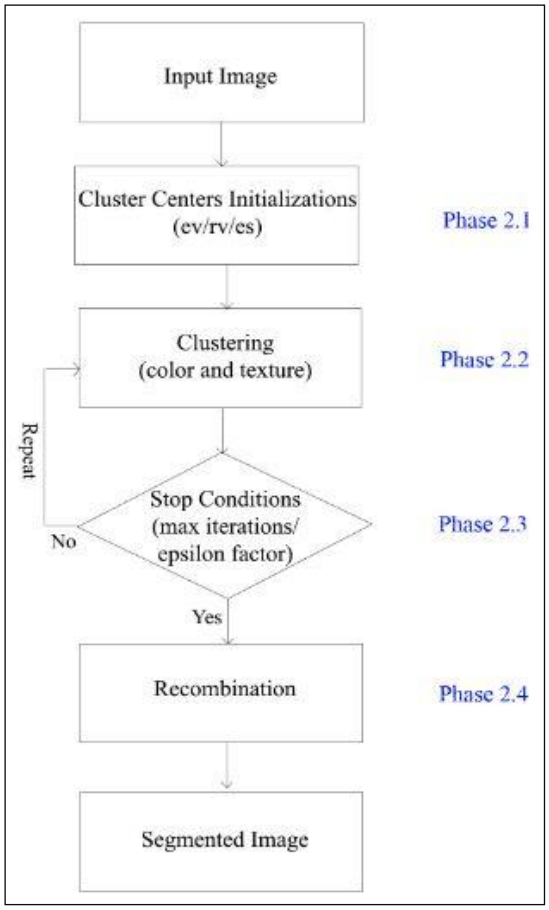
شکل ۱۰. نمونه خروجی تصویر بعد از اعمال تشخیص لبه شماره ۵، a) تصویر اصلی از راهرو دانشگاه دامغان، b) همان تصویر در سطح خاکستری، c) تصویر لبه‌یابی شده در راستای افقی، d) همان تصویر بعد از آستانه‌گذاری در راستای افقی، e) تصویر لبه‌یابی شده در راستای عمودی، f) همان تصویر بعد از آستانه‌گذاری در راستای عمودی

در الگوریتم K-means ابتدا k عضو (k تعداد خوشه‌ها است) بصورت تصادفی از میان n عضو به عنوان مراکز خوشه‌ها انتخاب می‌شود. سپس n-k عضو باقیمانده به نزدیک‌ترین خوشه تخصیص می‌یابند. بعد از تخصیص همه اعضا، مراکز خوشه مجدداً محاسبه می‌شوند و در تکرار بعدی، داده‌ها با توجه به مراکز جدید و محاسبه فاصله نزدیکی به مراکز خوشه‌ها، به هر خوشه تخصیص می‌یابند و این کار تا زمانی که مراکز خوشه‌ها ثابت شود، ادامه می‌یابد. بهترین خوشه‌بندی آن است که تشابه بین مرکز خوشه و همه اعضای همان خوشه را حداکثر و تشابه بین مراکز خوشه‌ها را حداقل کند. معادله ۱۰ نشان می‌دهد که چگونه الگوریتم K-means به طور معمول کار می‌کند.

$$\frac{1}{n}\sum_{i=1}^{n}\left[\min d^{2}(x_i, m_j)\right] \qquad (۱۰)$$

x، m، d^2 به ترتیب به نوع داده (پیکسل) میانگین مجموعه نقاط داده (مرکز خوشه‌ها) و حداقل فاصله اقلیدسی برمی گردد. می‌توان مراحل الگوریتم K-means را بصوت زیر نام برد:

۱-گروه بندی داده‌ها به k گروه، مقدار k از قبل مشخص شده است.
۲-k نقطه را بصورت تصادفی به عنوان نقاط مرکزی خوشه‌ها انتخاب شود.
۳-برای هر n-k نقطه، هرکدام به نزدیک‌ترین مرکز خوشه بر اساس تابع فاصله اقلیدسی اختصاص داده شود.
۴-نقطه مرکزی را به‌وسیله میانگین نقاط در هر خوشه محاسبه شود.
۵-مرحله ۳ و ۴ را تا زمانی که تمام نقاط خوشه‌بندی شوند و همچنین دیگر تغییری در اندازه و شکل خوشه‌ها به وجود نیاید ادامه داده شود.
نمای کلی معماری بخش‌بندی را در فلوچارت طراحی شده در شکل (۱۱) را مشاهده می‌فرمائید.



شکل ۱۱، طرح نمای کلی از تفکیک بر اساس خوشه‌بندی بر پایه K-means بهینه‌شده

تشخیص لبه به تنهایی برای استخراج مؤلفه‌های راهرو کافی نیست و نمی‌تواند ساختار راهرو و اشیایی که در آن وجود دارد را مشخص کند. پس در اینجا نیاز می‌بینیم که به کمک روش‌های دیگر به تکمیل روش تشخیص لبه بپردازیم تا به هدف مورد نظر که همان تولید مؤلفه‌های ساختارساز فضای راهرو است، برسیم. به وسیله خوشه‌بندی، تصویر به قسمت‌های مختلفی تقسیم می‌شود. هر یک از این قسمت‌ها با توجه به شباهت رنگ و بافت، پتانسیل نشان‌دادن یک شیء بامعنی در تصویر را دارند و علاوه بر آن خوشه‌بندی، به یافتن مؤلفه‌های ساختار راهرو می‌پردازد. در نتیجه ما به منظور قدرتمندسازی روش تشخیص لبه، آن را با یک الگوریتم تفکیک بر پایه خوشه-بندی K-means ادغام می‌کنیم.
روش K-means Clustering به وسیله اضافه کردن چند مؤلفه در رویکرد پیشنهادی این مقاله بهینه شده است. اولین بهبود مربوط به نقاط مرکزی خوشه‌ها هستند که در حالت معمول به صورت تصادفی در اولین تکرار انتخاب می شدند. در رویکرد معرفی‌شده از سه تکنیک برای انتخاب اولیه نقاط مرکزی استفاده می‌شود. سپس مابقی پیکسل‌ها براساس کمترین فاصله از نقاط مرکزی به خوشه‌ها اختصاص می‌یابند. دومین بهبود مربوط به فرایند خوشه‌یابی است که بطور معمول بر اساس محاسبه شباهت بر پایه ویژگی شدت روشنایی و رنگ انجام می‌شد که در این روش از فاکتور بافت نیز برای محاسبه شباهت میان اعضای هر خوشه هم استفاده می‌شود.
محسناتی که روش Optimal K-means clustering به نسبت K-means clustering ساده دارد را می‌توان در این سه مورد خلاصه کرد:
۱-پخش کردن نقاط اولیه با سه روش پیشنهاد شده
۲-استفاده از شباهت بافت علاوه بر شباهت رنگ به عنوان فاکتور دیگری برای خوشه‌بندی
۳-تعریف شروط توقف برای پایان دادن به فرآیند خوشه‌بندی، که این شرایط توقف را قبل از شروع خوشه‌بندی تعیین می‌کنیم. مؤلفه‌های بکار گرفته در شرایط توقف شامل تعداد تکرارها و عامل اپسیلون می باشد.

ورودی فلوچارت

بر طبق فلوچارت بالا در اولین مرحله از الگوریتم خوشه‌بندی، الگوریتم، ورودی از جنس تصویر RGB یا GrayScale یا سایر مدل‌های استاندارد را دریافت می‌کند.

فاز ۱,۲

تصویر با توجه به رنگ و بافتی که دارد ممکن است نیازمند نقاط اولیه متفاوتی باشد. در نتیجه روش‌های مختلفی برای

(a)
Smallest Color Value (Lower bound of the range) = (min r, min g, min b)

largest Color Value (upper bound of the range) = (max r, max g, max b)

Range = subtract (upper bound, lower bound)

(b)
Interval = range/(number of clusters -1)

درنظرگرفتن تمام ویژگی‌های سطحی و همچنین جلوگیری از بوجود آمدن اپتیمای محلی پیشنهاد شده است. این مقداردهی اولیه می‌تواند توسط نظر کاربر انتخاب شود.

۳ روش معرفی شده برای پیدا کردن نقاط اولیه مرکزی عبارت است از: الف) توزیع هم اندازه مقادیر در بین خوشه‌ها، ب) توزیع مقادیر بصورت تصادفی در بین خوشه‌ها، ج) توزیع هم اندازه نمونه‌ها در بین خوشه‌ها، که در ادامه به توضیح هرکدام خواهیم پرداخت.

الف) توزیع هم اندازه مقادیر در بین خوشه‌ها

اختصاص مقدار رنگ RGB پیکسل مرکزی از طریق تراورس کردن از دامنه پایین تا بالا براساس فاصله انجام می‌پذیرد. محدوده RGB برای نقاط مرکزی به وسیله فاصله‌ای که توسط اختلاف کوچکترین و بزرگترین رنگ به دست می‌آید محاسبه می‌شود (a). سپس فاصله نقاط مرکزی از هم نیز محاسبه می‌گردد (b) که در آن Smallest Color Value و largest Color Value به ترتیب مقدار کوچکترین و بزرگترین رنگ می‌باشند. همچنینRange محدوده رنگی و Interval فاصله است.

ب) توزیع مقادیر بصورت تصادفی در بین خوشه‌ها

مقدار RGB برای هر پیکسل مرکزی به وسیله انتخاب تصادفی از رنگ RGB موجود در تصویر انجام می‌گردد.

Central Point (R) = random (upper bound, lower bound)

Central Point (G) = random (upper bound, lower bound)

Central Point (B) = random (upper bound, lower bound)

پ) توزیع هم اندازه نمونه‌ها در بین خوشه‌ها

Evenly spaced samples distribution بر اساس مختصات x و y به دست می آید. زمانی که موقعیت یک نقطه مرکزی مشخص می‌شود مقدار رنگ RGB آن برابر با مقدار رنگ RGB پیکسلی که در آن موقعیت قرار دارد می‌شود.

توزیع نقاط مرکزی بر روی تصویر توسط یک فاصله بر حسب تعداد پیکسل‌ها انجام می‌پذیرد.

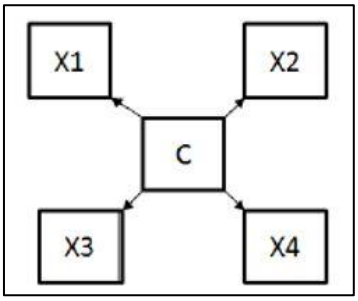
Total Number of Pixels = (height - 2) * (width - 2)

Interval = total number of pixels / number of clusters - 1

width نشان‌دهنده عرض و height نشان‌دهنده ارتفاع است و Total Number of Pixel تعداد کل پیکسل‌ها است.

فاز ۲,۲

در این مرحله برای جلوگیری از ابهامی که توسط ویژگی بافت ایجاد می‌شود، دو فاکتور مقدار رنگ RGB و Diagonal busyness factor برای خوشه‌بندی در نظرگرفته می‌شود. Diagonal busyness factor یا به اختصار DF خود شامل ۳ بخش عمودی، افقی و مورب است که به منظور احتمال وجود بافت در اطراف پیکسل بررسی می‌شود و نشان‌دهنده مجموع اختلاف رنگ بین پیکسل مرکزی و پیکسل‌های مورب است. به شکل ۱۲ برای تفهیم بیشتر مراجعه فرمائید.



شکل ۱۲. Diagonal busyness factor، جهت‌ها در اطراف یک پیکسل نشان داده شده است.

استدلال اینکه صرفاً از جهت مورب استفاده می‌شود و جهت‌های افقی و عمودی نادیده گرفته می‌شود این است که اگر جهت‌های دیگر را به آن اضافه کنیم، مقدار زیادی از بافت را در تصویر خواهیم داشت و تشخیص مناطق به عنوان شیء واقعی از بافت‌ها از هم دشوار خواهد شد.

فرمول DF را در ادامه مشاهده می‌کنید:

که می‌توان آن‌را به طور خلاصه به صورت معادله ۱۱ نمایش داد:

Let
c = central pixel
x_1 = upper left pixel
x_2 = upper right pixel
x_3 = bottom left pixel
x_4 = bottom right pixel
DF = diagonal factor

DF (R) = abs (c (r) – x_1 (r)) + abs (c (r) – x_2 (r)) + abs (c (r) – x_3 (r)) + abs (c (r) – x_4 (r))

DF (G) = abs (c (g) – x_1 (g)) + abs (c (g) – x_2 (g)) + abs (c (g) – x_3 (g)) + abs (c (g) – x_4 (g))

DF (B) = abs (c (b) – x_1 (b)) + abs (c (b) – x_2 (b)) + abs (c (b) – x_3 (b)) + abs (c (b) – x_4 (b))

$$DF = \sum_{i=1}^{4} |c - x_i| \qquad (۱۱)$$

C پیکسل مرکزی و x پیکسل‌هایی که در همسایگی پیکسل مرکزی هستند.

فاز ۳،۲ شرایط توقف

برای هر تکرار، دو نوع شرط توقف بررسی می‌شود و در صورت ملاقات، خوشه‌بندی خاتمه می‌یابد:

الف- حداکثر تکرارها: به معنای حداکثر بارهایی است که به طور متوسط فرآیند خوشه‌بندی باید انجام پذیرد.

ب- اپسیلون: مقدار اپسیلون مجموع فاصله اقلیدسی پیکسل‌ها متعلق به خوشه‌ای خاص از مرکز خوشه است.

فاکتور اپسیلون از بوجود آمدن خوشه‌های خیلی بزرگ و خیلی کوچک جلوگیری می‌کند. همچنین، ممکن است گاهاً با کاهش تعداد تکرارها همراه باشد. اگر هر کدام از شرایط ملاقات شود، فرآیند خوشه‌بندی متوقف می‌شود.

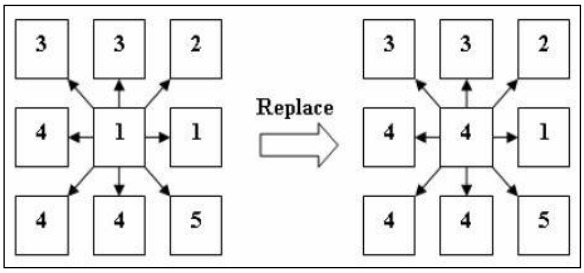
فاصله اقلیدسی به صورت معادله ۱۲ محاسبه می‌شود.

$$d(p,q) = \sqrt[2]{(x-s)^2 + (y-t)^2} \qquad (۱۲)$$

مختصات پیکسل p به صورت (x,y) و مختصات پیکسل q به صورت (s,t) نمایش داده می‌شود.

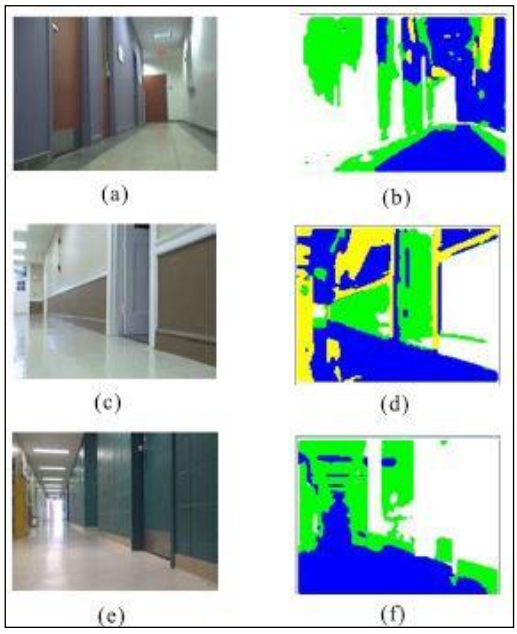
فاز ۴،۲ Recombination

از بازسازی هر پیکسل برای بالا بردن دقت خوشه‌بندی استفاده می‌شود. نحوه عملکرد آن به گونه‌ای است که تصویر از بالا به پایین تراورس می‌شوند و برای هر پیکسل، مقدار رنگ پیکسل را برابر با حداکثر تکرار یک مقدار خاص در همسایگی‌های اطراف پیکسل مرکزی قرار می‌دهد. نحوه عملکرد عملگر بازسازی را در شکل (۱۳) می‌بینید.



شکل ۱۳، نحوه عملکرد Recombination

در ادامه چند نمونه از تفکیک تصاویر از راهرو در دانشگاه دامغان و تز Yinxiao Li را مشاهده می‌کنید. در شکل (۱۴) مشاهده می‌شود که الگوریتم پیشنهادی ما با دقت عمل کرده و بخش‌بندی مناسبی را انجام داده است.



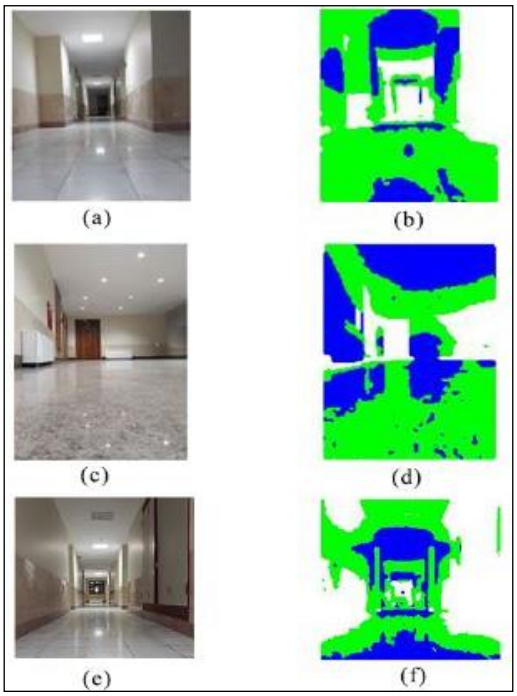
شکل ۱۴. نمونه‌ای از اعمال روش بخش‌بندی در کارهای پیشین، a, c, e)عکس اصلی از راهرو، b, d, f) تصویر تفکیک شده بوسیله خوشه‌بندی توسط Optimal K-means

در شکل (۱۵) بجز مناطقی که رفلکس آینه‌وار دارند، الگوریتم ما توانسته به تقسیم‌بندی تصویر بپردازد و درها و دیوارها را به درستی از کف تشخیص داده است. حتی با وجود بافت بر روی سطح راهرو، کل سطح راهرو را تقریباً به عنوان یک شیء در نظر گرفته است.

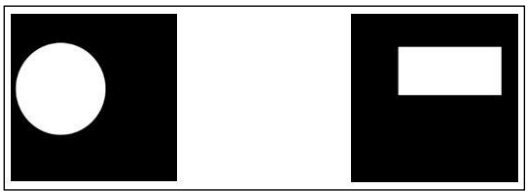
عملگرهای منطقی

عملگرهای منطقی برای انجام اعمال مشترک، اجتماع، اشتراک، تفریق بکار می‌رود که شاهد کاربردهای آن در ریاضیات، برق، کامپیوتر و دیگر علوم هستیم. در تصاویر دیجیتال نیز می‌توان از این عملگرها در راستای نیازهای خود استفاده کرد. البته با این شرط که این تصاویر از نوع

باینری باشند و پس‌زمینه با رنگ مشکی (۰) و پیش‌زمینه با رنگ سفید (۱) نشان داده شوند. دو تصویر را در شکل (۱۶) در نظر بگیرید.



شکل ۱۵، نمونه‌ای از اعمال روش بخش‌بندی از تصاویر راهرو دانشگاه دامغان، a, c, e عکس اصلی از راهرو، b, d, f) تصویر تفکیک شده بوسیله خوشه‌بندی توسط Optimal K-means



شکل ۱۶، تصویر چپ، تصویر A و تصویر سمت چپ، تصویر با نام B است.

عملگر XOR

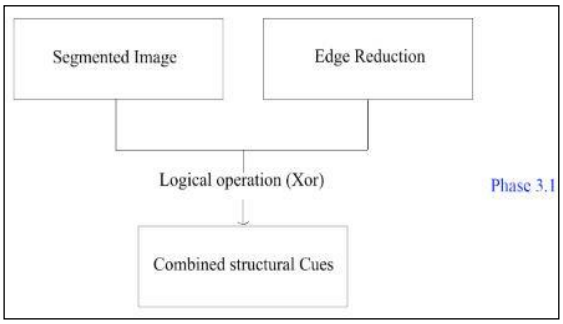
پیکسل‌هایی که به A یا B تعلق دارند و به هر دو آن‌ها متعلق نیستند، در حالت ریاضی به صورت A XOR B نشان داده‌می‌شود. شکل (۱۷) گویای عملکرد XOR است.

x	y	Output
1	1	0
1	0	1
0	1	1
0	0	0

شکل ۱۷، خروجی عملگر یا انحصاری بر دو تصویر A و B. (A XOR B)

۲-۳-فلوچارت ادغام دو روش تشخیص لبه و خوشه‌بندی به منظور تولید مولفه‌های ساختارساز فضا

در ادامه به بررسی چگونگی نحوه ترکیب دو فلوچارت قبلی می‌پردازیم. نمای کلی ترکیب دو فلوچارت تشخیص لبه در راستای دیوار-زمین و تفکیک بر اساس خوشه‌بندی بر پایه K-means بهینه‌شده در شکل (۱۸) را مشاهده می فرمائید.



شکل ۱۸. ترکیب خروجی‌های دو فلوچارت قبلی

فاز ۱,۳ استفاده از عملگرهای منطقی

در این فاز با استفاده از عملگرهای منطقی به ترکیب خروجی‌ها دو فلوچارت قبلی می‌پردازیم. نتیجه و خروجی بهتر زمانی حاصل می‌شود که این دو خروجی باهم ترکیب شوند. همانطور که از قبل گفته شد در رویکرد پیشنهاد شده، یک تصویر به عنوان ورودی دریافت می‌شود و به وسیله الگوریتم تشخیص لبه و آستانه‌گیری به پیدا کردن خطوط لبه پرداخته می‌شود. سپس، همان تصویر اولیه تحت یک روش بخش‌بندی بر پایه خوشه‌بندی قرار می‌گیرد و مناطق موجود در آن مشخص می‌شود. پس تا به اینجا دو نوع خروجی تصویر داریم: ا) تصویر لبه‌یابی شده ۲) تصویر بخش‌بندی شده.

حال به نام بردن برخی نامفهومی‌های پیش آمده در این رویکردها و رویکردهای مشابه و نحوه حل آن می‌پردازیم:

۱-ممکن است خطوط لبه نازک و شلوغ باشند: در این حالت از Histogram Equalization استفاده می‌کنیم تا با بالا بردن کنتراست خطوط لبه را پررنگ‌تر و ضخیم‌ترکنیم که در نهایت با استفاده از این روش و لبه‌یابی دوباره تصویر، خطوط لبه‌ی واضح و پررنگ‌تری داشته باشیم.

۲-ممکن است خطوط لبه در مرز دیوار-زمین هر کدام به تنهایی نشان‌دهنده مرزی از دیوار-زمین نباشد: در اینجا با

استفاده از عملگرهای منطقی(XOR، And ،Or) و اعمال بر روی هر دو تصویر لبه‌یابی سعی در پیدا کردن خطوط تلاقی که همان حد مرز دیوار-زمین باشد می‌پردازیم.

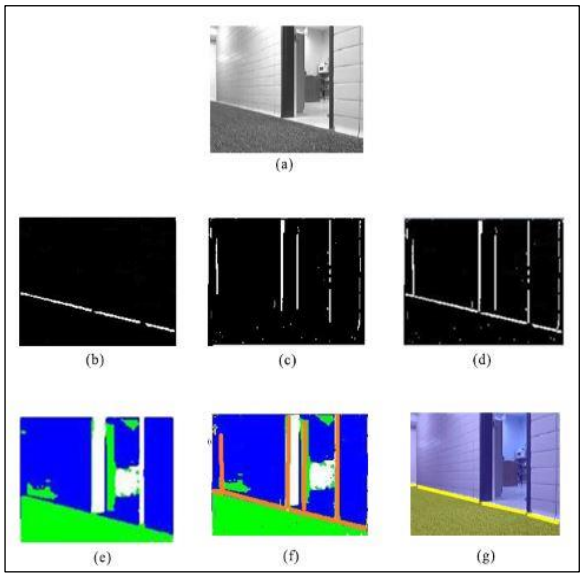
۳-ممکن است در عمل تفکیک و بخش‌بندی تصویر به دلیل تشابهات رنگی یا انعکاس‌هایی که در پایین دیوارها با سطح زمین رخ داده است، همپوشانی رنگی با سایر قسمت‌های راهرو در تصویر دیجیتال بوجود آمده باشد و قسمت‌هایی از دیوار به عنوان کف یا در حالت مخالف قسمت‌هایی از کف به عنوان دیوار در نظر گرفته شده باشد: در این صورت با استفاده از خروجی‌های لبه‌یابی شده از تصویر و تصویر بخش‌بندی شده، توسط یک عملگر منطقی و استفاده از هر دو خروجی تصویر به مشخص کردن حدود مرز دیوار و زمین می‌پردازیم.

۴-همچنین گاهی ممکن است لبه‌یابی به طور کامل انجام نشده باشد:

در اینجا نیز مانند شرایط بالا این‌بار با استفاده از خروجی تصویر بخش‌بندی شده به تصحیح لبه‌یابی می‌پردازیم.

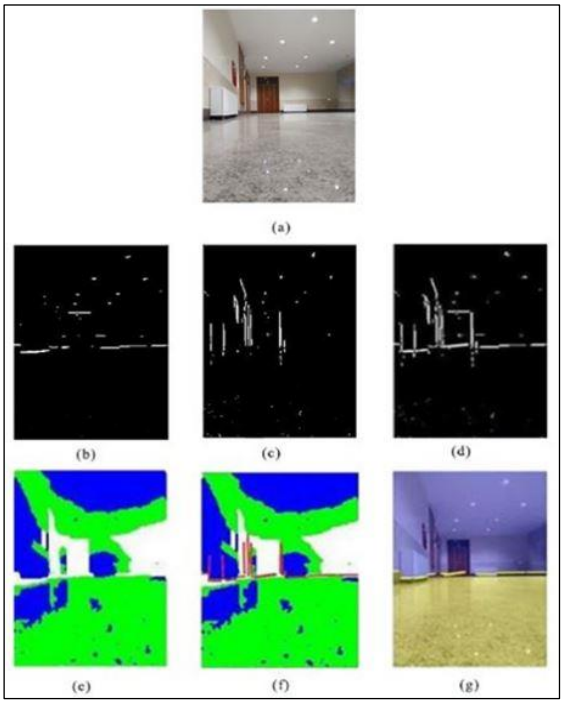
۳-نتایج تجربی

در شکل (۱۹) خروجی تصویر با رویکرد پیشنهادی و دقت بالای این رویکرد مشاهده می‌شود.



شکل ۱۹، نمونه خروجی تصویر راهرو a)تصویر سطح خاکستری، b)عکس لبه‌یابی شده در راستای افقی بعد از اعمال آستانه‌گذاری و کاهش خطوط لبه افقی بوسیله آستانه گیری توسط روش Iteration bisection، c)عکس لبه‌یابی شده در راستای عمودی بعد از اعمال آستانه‌گذاری و کاهش خطوط لبه عمودی بوسیله آستانه گیری توسط روش Iteration bisection، d)ترکیب خطوط لبه کاهش یافته شده افقی و عمودی توسط عملگر XOR، e)بخش‌بندی تصویر بر اساس روش Optimal K-means clustering، f)ترکیب تصویر خروجی لبه یابی شده و تصویر بخش‌بندی شده g)نمایش جدایی دیوار-زمین، محدوده دیوار به رنگ آبی و محدوده زمین به رنگ سبز و حد مرز جدایی دیوار و زمین به رنگ زرد نمایش داده شده است.

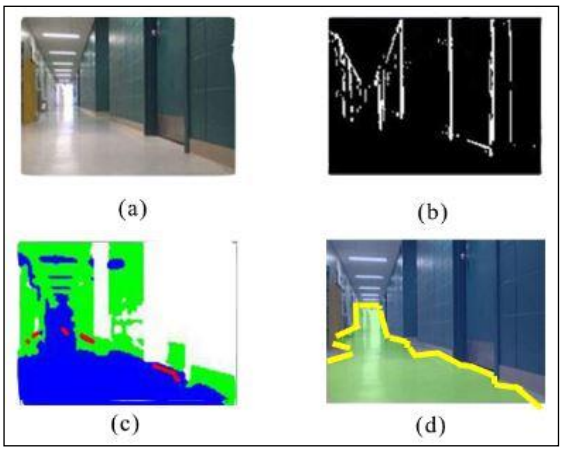
در شکل (۲۰) مشاهده می‌شود که در قسمت انتهایی راهرو به دلیل وجود سایه در پایین دیوار، الگوریتم تفکیک کمی دچار اشتباه شده است و دیوار را تا انتهای سایه‌ها ادامه داده است. اما همچنان این الگوریتم در قسمت‌های دیگر راهرو خوب عمل کرده و خروجی آن خوشه‌های با معنی است.



شکل ۲۰، نمونه خروجی تصویر راهرو دانشگاه دامغان a)تصویر اصلی، b)عکس لبه‌یابی شده در راستای افقی بعد از اعمال آستانه‌گذاری، c)عکس لبه‌یابی شده در راستای عمودی بعد از اعمال آستانه‌گذاری d) ترکیب خطوط لبه افقی و عمودی توسط عملگر XOR، e) بخش‌بندی تصویر بر اساس روش Optimal K-means clustering، f) ترکیب تصویر خروجی لبه یابی شده و تصویر بخش‌بندی شده ، g) نمایش جدایی دیوار-زمین. همانطور که مشاهده می‌کنید محدوده دیوار به رنگ آبی و محدوده زمین به رنگ سبز و حد مرز جدایی دیوار و زمین به رنگ زرد نمایش داده شده است.

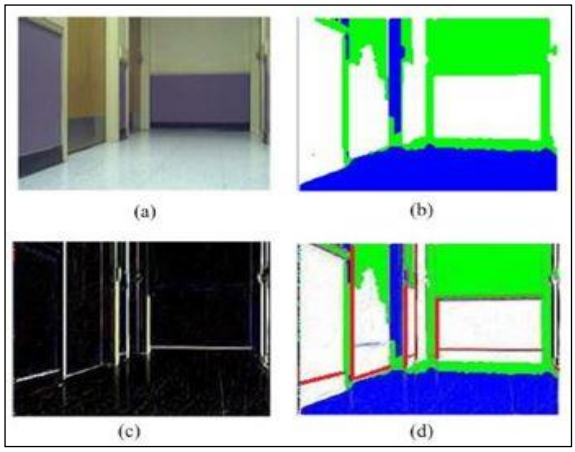
در خروجی‌های شکل (۲۱)، خطوط لبه افقی به درستی پیدا نشده است، به همین دلیل ما از نقاط تلاقی بین خطوط لبه افقی و عمودی استفاده کردیم و در نهایت مشاهده

می‌شود که حد مرز زمین و دیوار با ترکیب دو روش قبلی بهتر مشخص شده است.



شکل ۲۱، نمونه‌ای از ترکیب دو روش موجود در این مقاله، مشاهده می‌شود تفکیک دیوار و زمین به دلیل شباهت رنگی زمین و دیوار کمی سخت است. (a) تصویر اصلی از راهرو، (b) تصویر لبه‌یابی شده به همراه خطوط لبه افقی و عمودی کاهش یافته. برای اینکه لبه‌ها را نمایانتر شود از Histogram Equalization استفاده شد.(c) بخش‌بندی تصویر بر اساس Optimal K-means clustering، (d) تلفیق خطوط لبه و تصویر خوشه‌بندی شده، و به دست آوردن حد مرز پیوسته ای از کف و دیوارها، مرز با رنگ زرد نشان داده شده است.

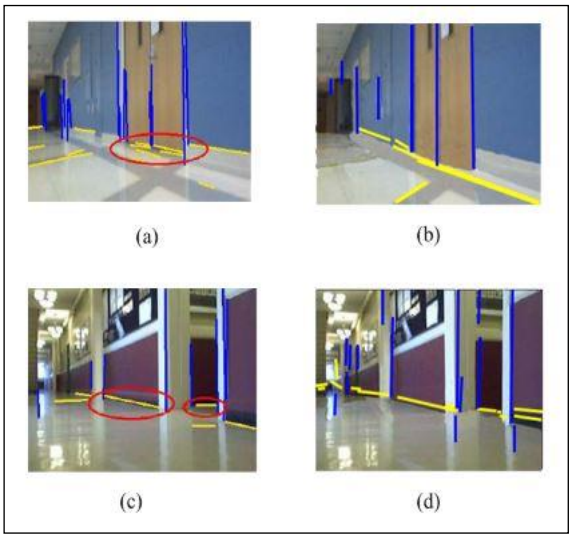
در شکل (۲۲) مشاهده می‌شود که در قسمت انتهایی راهرو به دلیل وجود سایه در پایین دیوار، الگوریتم تفکیک کمی دچار اشتباه شده است و دیوار را تا انتهای سایه‌ها ادامه داده است. اما همچنان این الگوریتم در قسمت‌های دیگر راهرو خوب عمل کرده و خوشه‌های خوبی به ما داده است.



شکل ۲۲، خروجی دیوار کف a)تصویر اصلی از راهرو، b) بخش‌بندی تصویر بر اساس روش Optimal K-means clustering ، c) ترکیب خطوط لبه افقی و عمودی توسط عملگر XOR ، d) تلفیق خطوط لبه و تصویر بخش‌بندی شده

۳-۱- مقایسه خروجی روش پیشنهادی در این رویکرد با رویکردهای پیشین

در شکل ۲۳، خروجی تصویر لبه‌یابی شده توسط روش پیشنهادی ما در مقایسه با روش پیشنهادی Yinxiao Li قرار گرفته است و مشاهده می‌شود که خطوط لبه در راستای افق و عمود بسیار دقیق و بهم پیوسته‌تر در روش پیشنهادی این مقاله به نسبت روش پیشنهادی Yinxiao Li شناسایی شده‌اند.



شکل ۲۳، مقایسه خروجی روش پیشنهادی در این پایان‌نامه با رویکردهای پیشین، تصویر (a) و (c) بوسیله Yinxiao Li لبه‌یابی شده است و (b) و (d) توسط روش پیشنهادی این مقاله، لبه‌یابی شده است.

در عمل درمی‌یابیم که در بعضی موارد بخش‌بندی سبب بهبود عملکرد تشخیص لبه می‌شود و در موارد دیگر استفاده از خطوط لبه سبب بهبود عملکرد بخش‌بندی می‌شود و اینگونه با ترکیب این دو خروجی و در بعضی موارد به کمک عملگر منطقی می‌توانیم به مرزبندی بهینه‌تری از قبل دست یابیم و همچنین در بیشتر مواقع بر درست تعیین کردن حد مرز دیوار-زمین مهر تائید بزنیم.

۴- نتیجه‌گیری و جمع‌بندی

ما در روش پیشنهادی در این مقاله سعی بر آن داشتیم که با استفاده از تکنیک‌های پردازش تصویر و به کمک بینایی کامپیوتر به تفکیک ساختارمندی از تصاویر راهرو برای هدایت ربات دست پیدا کنیم، که در این راه از روش‌ها و الگوریتم‌های مختلفی برای پیدا کردن خطوط لبه در تصویر و بخش‌بندی تصاویر استفاده کردیم. همچنین توانستیم تعدادی از نواقص موجود در روش‌های بکار گرفته شده در

رویکردهای پیشین را حل کنیم و در روال انجام و آزمودن روش خود به این نتیجه رسیدیم که با ترکیب خروجی‌های دو روش تشخیص لبه توسط سوبل و بخش‌بندی تصویر توسط روش K-means بهینه شده، می‌توانیم به ساختاری از فضای راهرو نظیر دیوار، کف و مرز دیوار-کف برای محدوده هدایت ربات دست یابیم.

از طرفی با مقایسه روش خود با روش‌های پیشین به این نتیجه رسیدیم که روش بکار گرفته‌شده، هم در تشخیص خطوط لبه و هم در بخش‌بندی به خوبی روش‌های پیشین و چه بسا موفق‌تر از روش‌های پیشین عمل کرده است. در نتیجه می‌تواند به عنوان پایه‌های مناسب در جهت حل مسائل مربوط به استخراج ساختار از فضای اطراف ربات برای هدایت در نظر گرفته شود.

مراجع

[۱] عباس نصر آبادی، ساسان آزادی وجواد حدادنیا، "آشکارسازی چهره انسان در تصاویر رنگی بر مبنای فیلتر گوسی"، نشریه مدل سازی در مهندسی، دوره۳، شماره ۱۷ ،۲۰۰۹.

[2] Ganesan, P., and G. Sajiv. "A comprehensive study of edge detection for image processing applications", IEEE International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), 2017, pp. 1-6.

[3] Li, Yinxiao, and Stanley T. Birchfield. "Image-based segmentation of indoor corridor floors for a mobile robot" IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 837-843.

[4] Gerogiannis, Demetrios, Christophoros Nikou, and Aristidis Likas. "Fast and efficient vanishing point detection in indoor images." In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), IEEE, 2012, pp. 3244-3247.

[5] Dhanachandra, Nameirakpam, and Yambem Jina Chanu. "A survey on image segmentation methods using clustering techniques", European Journal of Engineering and Technology Research, Vol. 2, NO. 1, 2017, pp. 15-20.

[6] Gupta, Surbhi, R. Sangeeta, Ravi Shankar Mishra, Gaurav Singal, Tapas Badal, and Deepak Garg. "Corridor segmentation for automatic robot navigation in indoor environment using edge devices", Computer Networks 178, 2020, pp. 107374.

[7] Canny, John. "A computational approach to edge detection", IEEE Transactions on pattern analysis and machine intelligence, (6), 1986, pp. 679-698.

[8] Kumar, BK Shreyamsha, "Image denoising based on gaussian/bilateral filter and its method noise thresholding", Signal, Image and Video Processing, Vol. 7, no. 6, 2013, pp. 1159-1172.

[9] Lohare, Deepali N., Ramesh R. Manza, and Neha Tiwari, "Comparative Study of Prewitt and Canny Edge Detector Using Image Processing Techniques", In Rising Threats in Expert Applications and Solutions, Singapore, 2021, pp. 705-713.

[10] Gupta, Samta, and Susmita Ghosh Mazumdar, "Sobel edge detection algorithm", International journal of computer science and management Research, Vol. 2, NO. 2, 2013, pp. 1578-1583.

[11] Shrivakshan, G. T., and Chandramouli Chandrasekar, "A comparison of various edge detection techniques used in image processing", International Journal of Computer Science Issues (IJCSI), Vol. 9, NO. 5, 2012, pp. 269.

[12] Buza, Emir, Amila Akagic, and Samir Omanovic, "Skin detection based on image color segmentation with histogram and k-means clustering", In 2017 10th International Conference on Electrical and Electronics Engineering (ELECO), 2017, pp. 1181-1186.

[13] Jain, Anil K, "Data clustering: 50 years beyond K-means", Pattern recognition letters, Vol. 31, NO. 8, 2010, pp. 651-666.

[14] Nayini, S. Ehsan Yasrebi, Somayeh Geravand, and Ali Maroosi, "A novel threshold-based clustering method to solve K-means weaknesses", In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017, pp. 47-52.

[15] Fahim, A. M., A. M. Salem, F. Af Torkey, and MA1101 Ramadan, "An efficient enhanced k-means clustering algorithm", Journal of Zhejiang University-Science A, Vol. 7, NO. 10, 2006, pp. 1626-1633.

[16] Felzenszwalb, Pedro F., and Daniel P. Huttenlocher, "Efficient graph-based image segmentation", International journal of computer vision, Vol.59, NO. 2, 2004, pp. 167-181.

[17] Bekhouche, Safia, and Yamina Mohamed Ben Ali, “Improvement of Quadree-Based Region Segmentation”, In 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), 2018, pp. 194-200. IEEE.

[18] Luo, Ming, Yu-Fei Ma, and Hong-Jiang Zhang, “A spatial constrained k-means approach to image segmentation”, In Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint, Vol. 2, 2003, pp. 738-742.

[9] Kumar, Suryansh, Ayush Dewan, and K. Madhava Krishna, “A bayes filter based adaptive floor segmentation with homography and appearance cues”, In Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, 2012, pp. 1-8.

[20] Magudeeswaran, V., C. G. Ravichandran, and P. Thirumurugan, "Brightness preserving bi-level fuzzy histogram equalization for MRI brain image contrast enhancement", International Journal of Imaging Systems and Technology, Vol. 27, NO. 2, 2017, pp. 153-161.

[۲۱] اسدی امیری و حمید حسن پور، "ارائه روشی برای پیش پردازش تصویر جهت بهبود عملکرد JPEG 2000 در فشرده سازی تصویر"، نشریه مدل سازی در مهندسی، دوره ۱۵، شماره ۴۸, ۲۰۱۷، صفحه ۲۴۷-۲۵۸.

[۲۲] راضیه راستگو و کورش کیانی، "شناسایی چهره بااستفاده از تنظیم دقیق شبکه های کانولوشنی عمیق و رویکرد یادگیری انتقالی"، نشریه مدل سازی در مهندسی، دوره ۱۷، شماره ۵۸, ۲۰۱۹، صفحه ۱۰۳-۱۱۱.