

بسم الله الرحمن الرحيم



دانشگاه حکیم بسزوری

دانشکده ریاضی و علوم کامپیوتر

پایان نامه برای دریافت درجه کارشناسی ارشد در رشته علوم تصمیم و مهندسی دانش

قطعه بندی تصاویر با روش درهم تنیدگی تصویر

استادان راهنما

دکتر محمود امین طوسی و دکتر مینا ملک زاده

پژوهشگر:

پروین بختی

تیر ۱۳۹۷



دانشگاه آزاد اسلامی

باسمه تعالی

فرم ارزشیابی و صورتجلسه دفاع از پایان نامه کارشناسی ارشد

فرم ۱۱۳-ت

جلسه دفاع از پایان نامه آقای /خانم پروین بختی دانشجوی رشته علوم تصمیم و مهندسی دانش به شماره دانشجویی ۹۴۱۳۱۳۷۰۱۷ با عنوان:

قطعه بندی تصاویر با روش درهم تنیدگی تصویر

در مورخه در دانشکده ریاضی و علوم کامپیوتر تشکیل و توسط هیات داوران مورد ارزشیابی قرار گرفت و نمره برابر درجه برای آن تعیین گردید .
به این ترتیب از این تاریخ آقای/ خانم پروین بختی به عنوان کارشناس ارشد در رشته مذکور شناخته می شود .

نمره کسب شده	حداکثر نمره	موارد	موارد ارزشیابی
	۴	رعایت اصول نگارش انسجام در تنظیم بخشهای مختلف، کیفیت تصاویر ، جداول و اشکال ،تنظیم فهرست ها ، منابع و ماخذ.	۱- کیفیت نگارش
	۱۰	بررسی تاریخچه و سابقه تجربی و نظری موضوع انسجام منطقی در بخش های مختلف پایان نامه ، ابتکار و نوآوری ، اهمیت و ارزش علمی پایان نامه ، استفاده از منابع معتبر و جدید ، کیفیت تجزیه و تحلیل یافته ها و نتیجه گیری ، روشن بودن روش کار ، هدف ها و فرضیه های تحقیق ، جدید بودن روش تحقیق	۲- کیفیت علمی
	۴	تسلط بر موضوع و بیان واضح و تفهیم آن ، توانایی در پاسخگویی به سوالات مطرح شده در جلسه ، رعایت زمان ارائه ، روش ارائه	۳- کیفیت ارائه در جلسه دفاع
	۱	گزارش های دوره ای پیشرفت کار (حداقل ۴ مورد)	۴- ارزشیابی گزارشات
	۱	مقاله مستخرج از پایان نامه: این نمره به صورت زیر اختصاص می یابد (۱) چکیده کنفرانسی هر مورد ۰/۲۵ نمره تا سقف ۰/۵ نمره (۲) مقاله کامل در مجموع مقالات همایشهای معتبر یا مقاله در مجلات علمی-ترویجی معتبر پذیرفته شده یا چاپ شده هر مورد ۰/۵ نمره تا سقف ۱نمره (۳)مقاله پذیرفته شده یا چاپ شده در مجلات علمی پژوهشی معتبر ۱ نمره (۴)مقاله ارسال شده به مجلات علمی پژوهشی معتبر هر مورد ۰/۲۵. نمره تا سقف ۰/۵ نمره (۵) دستگاه ساخته شده دارای گواهی ثبت اختراع یا به سفارش سازمان ها تا سقف ۱ نمره (۶) دستگاه ساخته شده کاربردی که به تأیید رئیس دانشکده رسیده باشد تا سقف ۰/۵ نمره	۵- خروجی پایان نامه
جمع			

درجه معادل کسب شده: (از ۱۹ تا ۲۰ عالی) از ۱۸ تا ۱۸/۹۹ بسیار خوب از ۱۶ تا ۱۷/۹۹ خوب از ۱۴ تا ۱۵/۹۹ قابل قبول کمتر از ۱۴ غیر قابل قبول

مشخصات هیات دوران

ردیف	نام و نام خانوادگی	سمت	مرتبۀ علمی	محل کار	امضا
۱	دکتر محمود امین طوسی	استاد راهنما	استادیار	دانشگاه حکیم سبزواری	
۲	دکتر مینا ملک زاده	استاد راهنما	استادیار	دانشگاه حکیم سبزواری	
۳	دکتر ...	استاد داور	استادیار	دانشگاه حکیم سبزواری	
۴	دکتر ...	نماینده تحصیلات تکمیلی	استادیار	دانشگاه حکیم سبزواری	

امضا

رئیس دانشکده

امضا

مدیر گروه



سوگند نامه دانش آموختگان دانشگاه حکیم سبزواری

به نام خداوند جان و خرد کزین برتر اندیشه بر نگذرد

اینک که به خواست آفریدگار پاک، کوشش خویش و بهره گیری از دانش استادان و سرمایه‌های مادی و معنوی این مرز و بوم، توشه‌ای از دانش و خرد گردآورده‌ام، در پیشگاه خداوند بزرگ سوگند یاد می‌کنم که در به کارگیری دانش خویش، همواره بر راه راست و درست گام بردارم. خداوند بزرگ، شما شاهدان، دانشجویان و دیگر حاضران را به عنوان داورانی امین گواه می‌گیرم که از همه دانش و توان خود برای گسترش مرزهای دانش بهره‌گیرم و از هیچ کوششی برای تبدیل جهان به جایی بهتر برای زیستن، دریغ نورزم. پیمان می‌بندم که همواره کرامت انسانی را در نظر داشته باشم و هموعان خود را در هر زمان و مکان تا سر حد امکان یاری دهم. سوگند می‌خورم که در به کارگیری دانش خویش به کاری که باره و رسم انسانی، آیین پرهیزگاری، شرافت و اصول اخلاقی برخاسته از ادیان بزرگ الهی، به ویژه دین مبین اسلام، مبادت دارد دست نیازم. همچنین در سایه اصول جهان شمول انسانی و اسلامی، پیمان می‌بندم از هیچ کوششی برای آبادانی و سرافرازی میهن و هم میهنانم فروگذاری نکنم و خداوند بزرگ را به یاری طلبم تا همواره در پیشگاه او و در برابر وجدان بیدار خویش و ملت سرافراز، بر این پیمان تا ابد استوار بمانم.

نام و نام خانوادگی: پروین بختی

تاریخ و امضا:

تأییدیه‌ی صحت و اصالت نتایج

باسمه تعالی

اینجانب پروین بختی به شماره دانشجویی ۹۴۱۳۱۳۷۰۱۷ دانشجوی رشته علوم تصمیم و مهندسی دانش مقطع تحصیلی کارشناسی ارشد تأیید می‌نمایم که کلیه‌ی نتایج این پایان‌نامه حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه برداری شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض در خصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می‌نمایم. در ضمن، مسئولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذی صلاح (اعم از اداری و قضایی) به عهده‌ی اینجانب خواهد بود و دانشگاه هیچ گونه مسئولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی: پروین بختی

تاریخ و امضا:

مجوز بهره برداری از پایان نامه

بهره برداری از این پایان نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر

تعیین می شود، بلامانع است:

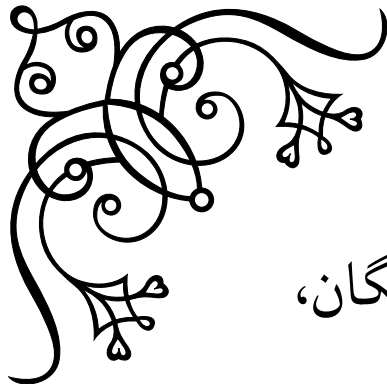
- بهره برداری از این پایان نامه برای همگان بلامانع است.
- بهره برداری از این پایان نامه با اخذ مجوز از استاد راهنما، بلامانع است.
- بهره برداری از این پایان نامه تا تاریخ ممنوع است.

استادان راهنما: دکتر محمود امین طوسی

دکتر مینا ملک زاده

تاریخ و امضا:

تقدیم به:



مهربان ترین فرشتگان، پدر و مادر عزیزم



شکر خداوندگار حکیم که هر چه طلب کردم از خدا بر منتهای همت خود کامران شدم. بسی شایسته است از استاد فرهیخته و فرزانه جناب آقای محمود امین طوسی که با کرامتی چون خورشید، سرزمین دل را روشنی بخشیدند و گلشن سرای علم و دانش را با راهنمایی های کار ساز و سازنده بارور ساختند؛ تقدیر و تشکر نمایم. و از پدر و مادر عزیز، دلسوز و مهربانم که آرامش روحی و آسایش فکری فراهم نمودند تا با حمایت های همه جانبه در محیطی مطلوب، مراتب تحصیلی و نیز پایان نامه درسی را به نحو احسن به اتمام برسانم؛ سپاسگزاری نمایم. همچنین، از بهترین همراه زندگیم که سایه مهربانیش سایه سار زندگیم می باشد، او که اسوه صبر و تحمل بوده و مشکلات مسیر را برایم تسهیل نمود سپاسگزاری می نمایم. و در نهایت، همه ی آنانی که در راه کسب دانش راهنمایم بودند و نفس خیرشان و دعای روح پرورشان بدرقه ی راهم بود. الهها به من کمک کن تا بتوانم ادای دین کنم و به خواسته ی همه شان جامه ی عمل بپوشانم.

پروین بختی

تیر ۱۳۹۷

فهرست مطالب

ج	فهرست جداول
د	فهرست تصاویر
۱	چکیده
۲	پیش‌گفتار
۴	فصل ۱: مقدمه و تاریخچه
۵	۱-۱ تعاریف
۹	۱-۱-۱ رگرسیون خطی
۱۳	۲-۱-۱ رگرسیون ريج
۱۴	۱-۲-۱-۱ نمایش اصلی رگرسیون ريج
۱۵	۲-۲-۱-۱ نمایش دوگان رگرسیون ريج
۱۶	۲-۱ نگاشت غیرخطی
۱۸	۳-۱ کارهای پیشین
۱۸	۱-۳-۱ درهم‌تنیدگی دودویی
۱۹	۲-۳-۱ روش درهم‌تنیدگی مبتنی بر یادگیری
۲۰	۳-۳-۱ الگوریتم روزان و توماسی
۲۱	۴-۳-۱ درهم‌تنیدگی مبتنی بر نظریه بیز
۲۳	۵-۳-۱ روش میثیما
۲۴	۶-۳-۱ روش حذفی
۲۴	۷-۳-۱ روش پواسن
۲۵	۸-۳-۱ روش فرم بسته
۲۶	۹-۳-۱ روش درهم‌تنیدگی سراسری

۳۰	فصل ۲: جستجوی نزدیکترین همسایگی
۳۰	۱-۲ نزدیکترین همسایگی
۳۴	۲-۲ درخت چندبعدی (KD-Tree)
۴۱	۱-۲-۲ اضافه کردن داده به درخت
۴۱	۲-۲-۲ حذف یک گره
۴۲	۳-۲-۲ جستجوی نزدیکترین همسایه

۵۱	فصل ۳: درهم تنیدگی با جستجوی نزدیکترین همسایگی
۵۱	۱-۳ درهم تنیدگی با جستجوی نزدیکترین همسایگی
۵۳	۱-۱-۳ محاسبه A با نزدیکترین همسایگی
۵۴	۲-۱-۳ بردار ویژگی x با مختصات مکانی
۵۵	۳-۱-۳ محاسبه تابع هسته برای تقسیم بندی نرم
۵۶	۴-۱-۳ پیاده سازی سریع برای روش فرم بسته
۵۸	۵-۱-۳ خلاصه ی روش

۵۹	فصل ۴: نتایج و پیشنهادات
۵۹	۱-۴ نتایج آزمایش
۶۲	۲-۴ بحث و نتیجه گیری
۶۲	۱-۲-۴ روش پیشنهادی
۶۳	۲-۲-۴ آزمایشات
۶۶	۳-۲-۴ نتیجه گیری
۶۶	۴-۲-۴ کارهای آتی

۶۷	فهرست منابع
----	-------------

۶۹	پیوست آ: برنامه های MATLAB روش های مطرح شده در پایان نامه
----	---

۸۱	واژه نامه فارسی به انگلیسی
----	----------------------------

۸۴	واژه نامه انگلیسی به فارسی
----	----------------------------

فهرست جداول

۲۳	نتایج سه روش درهم تنیدگی	۱-۱
۵۹	نتایج سه روش درهم تنیدگی	۱-۴
۶۰	زمان و میزان خطا برای سه الگوریتم درهم تنیدگی	۲-۴
۶۱	نتایج لایه های چندگانه درهم تنیدگی نزدیکترین همسایگی	۳-۴
۶۳	اطلاعات مربوط به مجموعه داده های استفاده شده در آزمایشات	۴-۴
۶۴	زمان اجرا و قتیکه ۳ داده انتخاب می شود	۵-۴
۶۴	میزان صحت و قتیکه ۳ داده انتخاب می شود	۶-۴
۶۴	زمان اجرا و قتیکه ۵ داده انتخاب می شود	۷-۴
۶۵	میزان صحت و قتیکه ۵ داده انتخاب می شود	۸-۴
۶۵	زمان اجرا و قتیکه ۱۰ داده انتخاب می شود	۹-۴
۶۵	میزان صحت و قتیکه ۱۰ داده انتخاب می شود	۱۰-۴

فهرست تصاویر

۲	نمونه عمل درهم‌تنیدگی تصویر	۱
۶	نقشه سه‌گانه	۱-۱
۱۰	نمودار پراکندگی	۲-۱
۱۹	نمونه عمل درهم‌تنیدگی تصویر	۳-۱
۲۰	نقشه سه‌گانه	۴-۱
۲۳	تصویر روش روزان و توماسی و روش مبتنی بر بییز	۵-۱
۲۴	تصاویر روش‌های نمونه‌گیری غیر پارامتری	۶-۱
۲۸	تصاویر نحوه کارکرد تابع کرنل	۷-۱
۳۲	کلاس طبقه‌بندی شده با ۱۵ همسایگی	۱-۲
۳۳	کلاس طبقه‌بندی شده با ۱ همسایگی	۲-۲
۴۱	نحوه ساخت درخت KD Tree	۳-۲
۴۳	الگوریتم مصور جستجو در درخت KD Tree	۴-۲
۵۰	نحوه جستجو در درخت KD Tree	۵-۲
۵۲	مقایسه روش‌های درهم‌تنیدگی فرم بسته، غیر محلی و نزدیکترین همسایگی	۱-۳
۵۴	مقایسه KNN و nonlocal	۲-۳
۵۴	محاسبات A در KNN matting	۳-۳
۵۶	مقایسه تقسیم بندی نرم و تقسیم بندی سخت	۴-۳



دانشگاه گیلان

فرم چکیده ی پایان نامه ی دوره ی تحصیلات تکمیلی

مدیریت تحصیلات تکمیلی

نام خانوادگی دانشجو: بختی	نام: پروین	ش. دانشجویی: ۹۴۱۳۱۳۷۰۱۷
استادان راهنما: دکتر محمود امین طوسی و دکتر مینا ملک زاده		
دانشکده ریاضی و علوم کامپیوتر	رشته: علوم تصمیم و مهندسی دانش	
مقطع: کارشناسی ارشد	تاریخ دفاع: تیر ۱۳۹۷	تعداد صفحات: ؟؟
عنوان پایان نامه: قطعه بندی تصاویر با روش درهم تنیدگی تصویر		
کلید واژه ها: درهم تنیدگی، یادگیری نیمه نظارتی، قطعه بندی، نزدیکترین همسایگی		
<p>چکیده: جداسازی دقیق پیش زمینه و پس زمینه یک مسئله مهم طبقه بندی یادگیری نیمه نظارتی است که مرتبط با موضوع پردازش تصویر می باشد. یادگیری نیمه نظارتی یعنی براساس محدودیت های ورودی کاربر مسئله مدلسازی می شود. یکی از روش های نسبتا جدید در این حوزه، روش های موسوم به درهم تنیدگی تصویر یا Image Matting می باشد که تحقیقات متعددی را به خود معطوف نموده است.</p> <p>در این تحقیق سعی بر آن است که به توصیف و تفسیر روش های روزان و توماسی، مبتنی بر بیز، میشیم، حذفی، هیستوگرام، مدل رنگ سراسری، پواسن و فرم بسته پرداخته شود. اساس کار این تحقیق توصیف و استفاده از روش درهم تنیدگی نزدیکترین همسایگی، به عنوان یک روش مطلوب تر نسبت به روش های توصیف شده از نظر زمان اجرا و میزان صحت، است که نتایج روش های متداول فرم بسته و مبتنی بر بیز با روش درهم تنیدگی نزدیکترین همسایگی مقایسه شده اند.</p> <p>در این تحقیق از روش درهم تنیدگی نزدیکترین همسایگی برای شناسایی اشیاء در تصویر به کار برده شده است و در نهایت از روش درهم تنیدگی جهت طبقه بندی داده های غیر تصویری استفاده شده است و نتایج مطلوبی بدست آمده است.</p> <p>در درهم تنیدگی تصاویر تصمیم گیری به صورت قطعی نیست و با یک احتمالی این عمل صورت می گیرد اما در مورد درهم تنیدگی در داده های غیر تصویر این عمل باید به صورت قطعی انجام شود که این خود باعث بروز مشکلاتی شد که در این تحقیق به حل این مشکلات پرداخته شده است.</p>		

پیش‌گفتار

استخراج اشیاء پیش‌زمینه از تصاویر نقش مهمی در ویرایش تصویر و ویدئو بازی می‌کند که به طور گسترده بیش از ۲۰ سال است که مورد مطالعه قرار گرفته است. یکی از مرسوم‌ترین و قدیمی‌ترین روش‌های مونتاز فیلم، استفاده از زمینه ثابت یا تکنیک پرده آبی است که اشیاء موردنظر را از یک تصویر برداشته و در یک زمینه دیگر قرار می‌دهند.

شکل ۱؟ یک نمونه از عمل درهم‌تنیدگی تصویر را برای استخراج شیء از زمینه نشان می‌دهد. در تصویر سمت راست تصویر یک گلدان را می‌بینید. روش درهم‌تنیدگی شیء گلدان را از پس زمینه جدا می‌کند که نتیجه را می‌توانید در تصویر سمت چپ ببینید. در این پایان‌نامه سعی بر این شده است که روش‌های درهم‌تنیدگی فرم بسته، مبتنی بر بیز و نزدیکترین



(ب) خروجی

(ا) تصویر اصلی

شکل ۱: نمونه عمل درهم‌تنیدگی تصویر

همسایگی تحلیل و تفسیر و با یکدیگر مقایسه شوند.

مقالات اصلی که طبق آن تحلیل‌های پایان‌نامه انجام شده است؛

1. Qifeng Chen, Dingzeyu Li and Tang, Chi-Keung. Knn matting. *IEEE Computer Society*, 35(13639689):2175 – 2188,2013
2. Wang, Jue and Cohen, Michael F. Image and video matting: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 97–175, 2008.

این پایان‌نامه شامل ۴ فصل است؛

در فصل ۱ این پایان‌نامه در ابتدا تعاریفی را جهت آشنایی با مطالب ارائه می‌دهیم سپس به توضیح مختصری از روش

های روزان و توماسی، مبتنی بر بیز، می‌شیم، حذفی، هیستوگرام، مدل رنگ سراسری، پواسن و فرم بسته می‌پردازیم؛

در فصل ۲ و ۳ روش درهم تنیدگی نزدیکترین همسایگی و الگوریتم KD-Tree جهت جستجو نزدیکترین همسایگی را شرح می‌دهیم؛
و در فصل ۴ مقایسه روش های فرم بسته و مبتنی بر بیز با روش نزدیکترین همسایگی انجام شده است. از درهم تنیدگی جهت یافتن اشیاء استفاده شده است و در نهایت با عمل درهم تنیدگی عمل طبقه بندی داده های غیر تصویری انجام شده است.

فصل ۱

مقدمه و تاریخچه

اگر عمل تقسیم بندی داده های ورودی به صورت با نظارت ^۲ باشد به آن طبقه بندی ^۳ می گویند. طبقه بندی یکی از مهمترین مباحث پایه ای در زمینه های آمار، یادگیری ماشین، شاسایی الگو و داده کاوی ^۴ می باشد. در الگوریتم های طبقه بندی مجموعه داده اولیه به دو مجموعه داده با عنوان مجموعه داده های آموزشی ^۵ و مجموعه داده های آزمایشی ^۶ تقسیم می شود، با استفاده از مجموعه داده های آموزشی مدل ساخته می شود و از مجموعه داده آزمایشی برای اعتبار سنجی و محاسبه دقت مدل ساخته شده، استفاده می شود. در الگوریتم های طبقه بندی چون ویژگی طبقه مربوط به هر رکورد مشخص است بنابراین جزء الگوریتم های با نظارت محسوب می شوند. الگوریتم های با نظارت شامل دو مرحله با عنوان آموزش (یادگیری) و مرحله ارزیابی هستند [۴]. روش درهم تنیدگی ^۷ یک مسئله یادگیری ماشین است که هدف قطعه بندی دقیق تصویر به دو لایه پیش زمینه و پس زمینه است. اگر میزان شدت روشنایی یک پیکسل را I ، میزان مشارکت تصویر پیش زمینه در تصویر اصلی را F و میزان مشارکت تصویر پس زمینه در تصویر اصلی را B بنامیم؛ آنگاه هدف، بدست آوردن میزان α به نحوی است که ترکیب خطی I مطابق معادله α بدست آید.

$$I = \alpha F + (1 - \alpha)B \quad (1-1)$$

با توجه به داشتن تنها یک تصویر، سه مقدار F ، B ، α مقادیری نامعلوم هستند و باید این سه مقدار در هر پیکسل معین شوند. در تصاویر خاکستری مدل رنگ یک بعدی است و بنابراین برای هر پیکسل باید یک مقدار را بدست آوریم که طبق معادله α مقدار نامعلوم وجود دارد. این در صورتی است که تنها یک معادله وجود دارد. این مسئله در مدل رنگ سه بعدی (RGB) ^۸ به این صورت است که برای هر پیکسل باید سه مقدار را بدست بیاوریم؛ که با توجه به اینکه برای هر پیکسل سه لایه موجود می باشد برای هر پیکسل α مقدار نامعلوم وجود دارد. این در صورتی است که تنها سه معادله وجود دارد. در هر دو مدل ذکر شده، در سیستم معادلات بی نهایت جواب موجود می باشد.

^۱Supervised Method ^۲Classification ^۴Data Mining ^۵Train Dataset ^۶Test Dataset

^۷Image Matting ^۸Red,Green,Blue

۱-۱ اصطلاحات و مفاهیم

کوچکترین جزء هر تصویر پیکسل (pixel) است. پیکسل مخفف picture element به معنای المان تصویر است. یک تصویر متشکل از تعداد زیادی پیکسل است که در کنار یکدیگر قرار گرفته و هرکدام از این نقاط دارای مکان و مقدار مشخصی هستند [۴].

برای نمایش یک تصویر $M \times N$ از یک ماتریس که M سطر و N ستون دارد، استفاده می‌شود. مقدار هر درایه نشان دهنده‌ی شدت روشنایی تصویر در آن نقطه است. هر عنصر می‌تواند مقداری بین ۰ و ۲۵۵ داشته باشد. مقدار صفر نشان دهنده‌ی رنگ سیاه و ۲۵۵ نشان دهنده‌ی رنگ سفید می‌باشد.

یک تصویر RGB متشکل از سه ماتریس است که هر یک از آن‌ها مقادیر قرمز (Red) و سبز (Green) و آبی (Blue) تصویر رنگی را نگه می‌دارند. یک پیکسل زمانی مقدار خاکستری خواهد داشت که مؤلفه‌های R, G, B آن مقدار یکسانی داشته باشند.

معمولاً از لغت خاکستری^۱ در ارتباط با شدت روشنایی^۲ تصاویر تک رنگ استفاده می‌شود. تصاویر رنگی از ترکیب تصاویر تک رنگ تشکیل می‌شوند. به عنوان مثال در سیستم رنگی RGB، تصویر رنگی از سه تصویر (قرمز، سبز و آبی) تشکیل می‌شود.

آستانه‌گیری^۳: هنگامی که قرار است از یک تصویر رنگی تصویر سیاه و سفید ساخته شود، برای این کار باید تک تک پیکسل‌ها با مقدار مشخصی مقایسه شده اگر بزرگ تر از آن مقدار بود به جای آن رنگ سفید و اگر کوچک تر بود به جای آن رنگ سیاه قرار می‌گیرد.

هیستوگرام تصویر: هیستوگرام در واقع همان نمودار میله‌ای افزونی هر رنگ می‌باشد. از هیستوگرام برای آستانه‌گیری استفاده می‌شود.

بیش برآزش^۴: به پدیده‌ی نامطلوبی در آمار گفته می‌شود که در آن درجه آزادی مدل بیشتر از درجه آزادی واقعی انتخاب شده و در نتیجه اگر چه مدل روی داده‌های استفاده شده برای یادگیری بسیار خوب نتیجه می‌دهد، اما بر روی داده جدید دارای خطای زیادی است. منظم سازی^۵ از راه‌های مقابله با این پدیده است.

اعتبار سنجی متقابل^۶: در این نوع اعتبار سنجی، داده‌ها به K زیر مجموعه افزای می‌شوند. از این K زیر مجموعه، هر بار یکی برای اعتبار سنجی و $K - 1$ تای دیگر برای مدلسازی به کار می‌روند. این روال K بار تکرار می‌شود و همه داده‌ها دقیقاً یک بار برای مدلسازی و یک بار برای اعتبار سنجی به کار می‌روند. در نهایت میانگین نتیجه این K بار اعتبار سنجی به عنوان یک تخمین نهایی برگزیده می‌شود. به طور معمول از $K = 10$ استفاده می‌شود.

نقشه سه گانه^۷: کاربر هر تصویر را به سه بخش تقسیم می‌کند. این تقسیم بندی در شکل؟؟ نشان داده شده است:

R_f : پس زمینه

R_b : پیش زمینه

^۱Gray ^۲Intensity ^۳Threshold ^۴Overfeating ^۵Regularization ^۶Cross Validation

^۷Trimap



(ب) نقشه سه گانه درشت



(آ) نقشه سه گانه ریز



(د) نقشه سه گانه رنگ-پیکسل



(ج) نقشه سه گانه رنگی

شکل ۱-۱: نقشه سه گانه

R_u : بدون برچسب (ناشناخته)

درواقع این سه قسمت به عنوان نقشه سه گانه^۱ شناخته شده است. نقشه سه گانه می تواند انواع متفاوتی داشته باشد. مدل های آن در تصویر؟؟ نشان داده شده است. نقشه سه گانه ریز و درشت به صورت سیاه و سفید هستند. یعنی پیش زمینه با رنگ سفید پس زمینه با رنگ سیاه و منطقه ناشناخته با رنگ خاکستری مشخص شده است. فرقیان این است که محدوده ناشناخته در نقشه سه گانه درشت بیشتر از محدوده ناشناخته نقشه سه گانه ریز است. نقشه سه گانه رنگی و نقشه سه گانه رنگ-پیکسل به صورت یک تصویر رنگی هستند. در مورد نقشه سه گانه رنگی پیش زمینه با رنگ سفید، پس زمینه با رنگ سیاه و منطقه ناشناخته به همان صورتی که در تصویر اصلی است، باقی می ماند. در مورد نقشه سه گانه رنگ-پیکسل، پیش زمینه با چند پیکسل رنگی و پس زمینه با چند پیکسل رنگی دیگر نشانه گذاری می شوند و منطقه ناشناخته به همان صورت رنگی که در تصویر اصلی است، باقی می ماند.

FLANN: یک کتابخانه برای انجام جستجوی تقریبی سریع نزدیکترین همسایگی در ابعاد بالا است. FLANN در

C++ نوشته شده است و قابل پیوند با زبان های C و MATLAB و python است.

تجزیه و ررونوی: در علم ریاضیات تجزیه و ررونوی روشی برای تقسیم فضا به تعدادی ناحیه می باشد. در این تجزیه به هر مجموعه ای از نقاط ناحیه ای اختصاص داده می شود. این نواحی سلول های ورونوی نامیده می شود. برای یک مجموعه از نقاط، تجزیه و ررونوی سطح را به مناطقی تقسیم بندی می کند که برای هر نقطه از مجموعه نقاط یک منطقه تعریف می شود. به طوری که تمام نقاط این منطقه به نقطه تولید کننده آن منطقه نزدیکتر می باشد.

^۱Trimap

گرادیان مزدوج^۲: روش گرادیان مزدوج یا روش گرادیان همیوگ در ریاضیات، الگوریتمی برای حل سیستم معادلات خطی می‌باشد. معادلاتی که ماتریس آنها متقارن و مثبت معین است. این روش از نوع الگوریتم‌های تکراری می‌باشد. pcg^۱: یک تکنیک جهت حل معادلات خطی $A * x = b$ با روش گرادیان مزدوج است، که A یک ماتریس $N \times N$ است.

نکته: منظور از $\operatorname{argmin}_x f(x)$ پیدا کردن مقدار x است به طوری که به ازای آن تابع مینیم شود. به طور مثال

$$\min_x [x^2 + 1] = 1$$

در حالی که

$$\operatorname{argmin}_x [x^2 + 1] = 0$$

می‌باشد.

۲-۱ تعاریف و قضایا

تعریف ۱. فرض کنید A یک ماتریس $m \times n$ باشد. ترانهاد^۲ ماتریس A با نماد A^T نمایش داده می‌شود، که یک ماتریس $n \times m$ است که به صورت زیر تعریف می‌شود.

$$A = (a_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \iff A^T = (a_{ji})_{\substack{1 \leq j \leq n \\ 1 \leq i \leq m}}$$

قضیه ۱. فرض کنید T عملگر ترانهاد ماتریس باشد:

$$T : M_{m \times n} \longrightarrow M_{n \times m}$$

$$A \longrightarrow A^T$$

آن‌گاه روابط زیر برقرار می‌باشند [۱]:

$$A^{TT} = A \bullet$$

$$(A + B)^T = A^T + B^T \bullet$$

$$(AB)^T = B^T A^T \bullet \quad ; \quad \text{اگر } n = m \text{ باشد.}$$

^۲ Conjugate Gradient Methods

^۱ Preconditioned Conjugate Gradient

^۲ Transpose

• $(A^{-1})^T = (A^T)^{-1}$ ؛ اگر A معکوس پذیر و $n = m$ باشد.

تعریف ۲. $\|\cdot\|_1$ نرم L_1 در \mathbb{R}^n است و به صورت زیر تعریف می شود:

$$\forall X = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n; \quad \|X\|_1 = \sum_{k=1}^n |x_k|$$

تعریف ۳. $\|\cdot\|_2$ نرم L_2 در \mathbb{R}^n است و به صورت زیر تعریف می شود:

$$\forall X = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n; \quad \|X\|_2 = \sqrt{X^T X} = \sqrt{\sum_{k=1}^n x_k^2}$$

تعریف ۴. اگر $f: \mathbb{R}^n \rightarrow \mathbb{R}$ ، تابعی مشتق پذیر باشد، آن گاه گرادیان تابع به صورت زیر تعریف می شود:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)^T \in \mathbb{R}^{n \times 1}$$

و اگر $f = (f_1, f_2, \dots, f_m): \mathbb{R}^n \rightarrow \mathbb{R}^m$ مشتق پذیر باشد، آن گاه

$$\nabla f = \frac{\partial (f_1, f_2, \dots, f_m)}{\partial (x_1, x_2, \dots, x_n)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \in \mathbb{R}^{m \times n}. \quad (2-1)$$

قضیه ۲. فرض کنید $f, g: \mathbb{R}^n \rightarrow \mathbb{R}^m$ دو تابع مشتق پذیر به صورت $f = (f_1, f_2, \dots, f_m)^T$ و $g = (g_1, g_2, \dots, g_m)$ باشند، و تابع $h: \mathbb{R}^n \rightarrow \mathbb{R}$ به صورت $h(x) = f(x)^T g(x)$ تعریف شده باشد، آن گاه تابع h مشتق پذیر است و مشتق آن از رابطه زیر به دست می آید:

$$\nabla h(x) = \nabla f(x)^T g(x) + \nabla g(x)^T f(x)$$

تعریف ۵. فرض می کنیم ماتریس $A \in \mathbb{R}^{m \times n}$ باشد. آن گاه ماتریس $A^\dagger \in \mathbb{R}^{n \times m}$ شبه معکوس^۱ ماتریس A است، اگر $AA^\dagger A = A$ و ماتریس های $U \in \mathbb{R}^{n \times n}$ و $V \in \mathbb{R}^{m \times m}$ موجود باشند به طوری که

$$A^\dagger = UA^T$$

$$A^\dagger = A^T V$$

^۱Pseudo-inverse

برقرار باشند.

شرایط لازم $A^\dagger = UA^T = A^T V$ را می‌توان به صورت زیر تعبیر نمود. هر سطر از ماتریس شبه معکوس A^\dagger از A ترکیب خطی از سطرها A^T است و هر ستون A^\dagger ترکیب خطی از ستون‌های A^T است.

قضیه ۳. فرض می‌کنیم ماتریس $A \in \mathbb{R}^{m \times n}$ و $m \geq n$ و $rank(A) = n$ باشد، آن‌گاه شبه معکوس A به صورت $A^\dagger = (A^T A)^{-1} A^T$ حاصل می‌شود.

برهان. اولاً $A(A^T A)^{-1} A^T A = A$ است. اگر $U = (A^T A)^{-1}$ و $V = A(A^T A)^{-1} A^T$ قرار دهیم، آن‌گاه $A^\dagger = UA^T = A^T V$ می‌باشد. در حقیقت

$$A^\dagger A = I$$

می‌باشد.

قضیه ۴. فرض می‌کنیم ماتریس $A \in \mathbb{R}^{m \times n}$ و $m \leq n$ و $rank(A) = m$ باشد، آن‌گاه شبه معکوس A به صورت $A^\dagger = A^T (AA^T)^{-1}$ حاصل می‌شود.

۱-۲-۱ رگرسیون خطی

مجموعه آموزشی^۱ داده شده $S = \{(x_1, y_1), \dots, (x_l, y_l)\}$ از نقاط x_i از $X \subseteq \mathbb{R}^n$ با برچسب‌های نظیر به نظیر y_i در $Y \subseteq \mathbb{R}$ را در نظر می‌گیریم. هدف یافتن یک تابع خطی مانند $g(\cdot)$ به صورت زیر است:

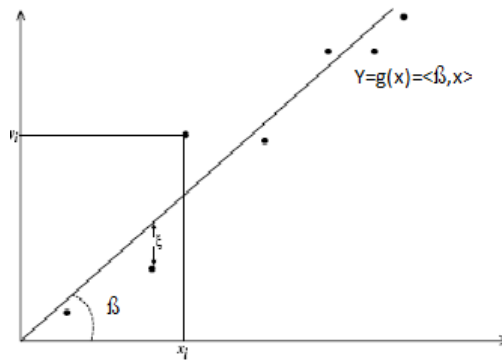
$$g(x) = \langle \beta, x \rangle = \beta^T = \sum_{i=1}^n \beta_i x_i$$

و ایجاد یک تابع الگو که باید تقریباً معادل صفر باشد:

$$f(x, y) = |y - g(x)| = |y - \langle \beta, x \rangle| \approx 0.$$

در واقع دنبال یافتن β می‌هستیم که در روابط فوق صدق کند، که بهترین درونیایی مجموعه آموزشی داده شده S است. این به طور طبیعی یکی از ساده‌ترین روابطی است که می‌تواند در مبدا $X \times Y$ یافت شود، یعنی یک تابع خطی g از X مقدار نظیر به نظیر Y را تنظیم می‌کند. از لحاظ هندسی متناظر ساختن به منظور برازش یک ابر صفحه از میان نقاط داده n -بعدی می‌باشد، که از نمادگذاری $x = (x_1, x_2, \dots, x_n)^T$ برای بردارهای داده n بعدی استفاده می‌کنیم. شکل $g(x)$ ؟ یک مثال برای $n = 1$ را نشان می‌دهد. در نمونه دقیق، هنگامی که داده‌ها به صورت $(x, g(x))$ تولید شده باشند که

^۱Training set



شکل ۱-۲: نمودار پراکندگی

زیر حاصل می‌شود: $g(x) = \langle \beta, x \rangle$ و دقیقاً $l = n$ نقطه مستقل خطی وجود داشته باشد، پارامتر β به وسیله حل سیستم معادلات خطی

$$X\beta = Y.$$

که از نماد X برای نمایش ماتریسی که سطرهاى آن بردارهای سطرى x_1^T, \dots, x_l^T و Y برای نمایش بردار $(y_1, \dots, y_l)^T$ استفاده می‌شود. در واقع داده‌ها بردارهای ستونی اند اما در ماتریس X به عنوان بردارهای سطرى ذخیره می‌شوند. اگر تعداد نقاط کمتر از بعد باشد، ممکن است چندین β وجود داشته باشد که دقیقاً داده‌ها را توصیف کند. یک معیار برای انتخاب بین آن‌ها لازم است، که در این حالت β با نرم کمینه مطلوب است. اگر تعداد نقاط بیشتر از بعد باشد و در فرآیند تولید داده‌ها، نویز وجود داشته باشد پس نمی‌توان انتظار داشت الگو دقیق باشد بنابراین معیاری لازم است. در این وضعیت الگویی با کمترین خطا را انتخاب می‌کنیم.

به طور کلی اگر داده‌ها همراه با خطا باشند ترکیبی از دو استراتژی به کار گرفته می‌شود: یافتن بردار β ای با کمترین نرم و کمترین خطا. در شکل؟؟ فاصله نشان داده شده با ξ نشان دهنده خطای تابع خطی در مثال آموزشی است، یعنی $\xi = y - g(x)$ بنابراین

$$f(x, y) = |y - g(x)| = |\xi|$$

می‌باشد. هدف یافتن تابعی است که مجموع خطاها روی داده‌های آموزشی کمینه باشد.

مجموع مربعات خطا رایج ترین معیار انتخاب اختلاف بین داده‌های آموزشی و تابع مورد نظر ($g(\cdot)$) است:

$$L(g, s) = L(g, \beta) = \sum_{i=1}^l (y_i - g(x_i))^2 = \sum_{i=1}^l \xi_i^2 = \sum_{i=1}^l L((x_i, y_i), g)$$

از نمادگذاری $L((x_i, y_i), g) = \xi_i^2$ برای نمایش مربعات خطا یا اختلاف g در نمونه (x_i, y_i) و از نماد $L(f, S)$ برای نمایش مجموع اختلاف تابع f بر مجموعه‌ی آموزشی S استفاده می‌شود. مسئله یادگیری اکنون به برآورد بردار β که کمینه

خطا را دارد، تبدیل می‌شود. این مسئله توسط گاوس^۱ و با عنوان تقریب کمترین مربعات^۲ معرفی شد. با به کارگیری نمادگذاری بالا، بردار اختلاف خروجی را می‌توان به صورت زیر نوشت:

$$\xi = Y - X\beta$$

بنابراین تابع خطا^۱ به صورت زیر حاصل می‌شود:

$$L(\beta, s) = \|\xi\|_2^2 = (Y - X\beta)^T(Y - X\beta) \quad (3-1)$$

مقدار بهینه β به وسیله مشتق‌گیری از تابع خطا نسبت به پارامتر β و برابر صفر قرار دادن آن، تعیین می‌شود:

$$\frac{\partial L(\beta, s)}{\partial \beta} = -2X^T Y + 2X^T X\beta = 0$$

بنابراین معادلات زیر حاصل می‌شوند که به آن معادلات نرمال^۲ می‌گویند.

$$X^T X\beta = X^T Y \quad (4-1)$$

اگر $X^T X$ معکوس پذیر^۳ باشد، جواب مسئله کمترین مربعات به صورت زیر حاصل می‌شود:

$$\beta = (X^T X)^{-1} X^T Y$$

پیش بینی خروجی برای نقاط داده جدید می‌تواند با به کارگیری تابع پیش بینی محاسبه شود: $g(x) = \langle \beta, x \rangle$

ملاحظه ۱. (نمایش دوگان^۴). اگر معکوس $X^T X$ موجود باشد می‌توان β را به روش زیر نیز بیان کرد:

$$\beta = (X^T X)^{-1} X^T Y = X^T X (X^T X)^{-2} X^T Y = X^T \alpha$$

که در آن

$$\alpha = X (X^T X)^{-2} X^T Y$$

می‌باشد.

ملاحظه ۲. (شبه معکوس). اگر $X^T X$ منفرد^۵ (معکوس ناپذیر) باشد، در این حالت برای یافتن جواب معادله؟؟ از

^۱Gauss ^۲Least squares approximation ^۳Normal Equations ^۴Invertible

^۵Dual representation ^۵Singular

شبه معکوس استفاده می‌کنیم.

تعریف ۶. دستگاه معادلات خطی زیر را در نظر می‌گیریم:

$$A_{n \times n} x_{n \times 1} = b_{n \times 1}$$

با فرض اینکه A غیر منفرد باشد، خواهیم داشت:

$$x = A^{-1}b$$

حال اگر b شامل نویز یا خطای محاسباتی ناشی از گرد کردن مانند Δb باشد، در این صورت این خطا به صورت زیر در پاسخ ظاهر خواهد شد:

$$x + \Delta x = A^{-1}(b + \Delta b)$$

لذا می‌توان نوشت:

$$\Delta x = A^{-1}\Delta b$$

با توجه به خواص نرم ماتریس‌ها نتیجه می‌شود؟؟:

$$\|\Delta x\|_2 \leq \|A^{-1}\|_2 \|\Delta b\|_2$$

از عبارت اخیر می‌توان تعبیر کرد که، اگر $\|A^{-1}\|_2$ مقدار کوچکی باشد، برای تغییرات کم در b یعنی $\|\Delta b\|_2$ کوچک، مقدار $\|\Delta x\|_2$ کم خواهد بود. ولی برای $\|A^{-1}\|_2$ های بزرگ، مقدار $\|\Delta x\|_2$ بزرگ است، حتی اگر $\|\Delta b\|_2$ مقدار کوچکی باشد.

لذا برای تشخیص بد حالت بودن یک دستگاه معادلات، پارامتری به نام عدد حالت تعریف می‌گردد:

$$\kappa = \|A\|_2 \|A^{-1}\|_2 \geq 1$$

اگر عدد حالت کوچک باشد، بیان‌کننده آن است که ماتریس A و دستگاه معادلات حاصل خوش حالت (خوش وضع^۱) است و اگر عدد حالت مقدار خیلی بزرگی باشد، بیان‌گر آن است که ماتریس نزدیک به منفرد شدن است، لذا آن ماتریس را بدحالت (بد وضع^۲) نامیده و خطای محاسباتی در معکوس کردن ماتریس A زیاد است.

تعریف ۷. مسائل بد طرح: در اوایل قرن پیش، هادامارد^۳ اولین کسی بود که بحث دستگاه‌های معادلات خوش طرح^۴

^۱Well-conditioned

^۲Ill-conditioned

^۳Hadamard

^۴Well-posed

و بد طرح^۵ را مطرح نمود. ایشان دستگاه معادلاتی را که در آن ها سه شرط "وجود، یکتایی و پایداری جواب" تأمین باشند، را مسائل خوش طرح و مسائلی که در آن ها یکی از سه شرط نقض گردیده باشد، را مسائل بد طرح نام گذاری کردند.

در بسیاری از عملیات جبری، لازم است یک تقریب مناسب مانند \hat{x} از بردار $x \in \mathbf{R}^n$ به دست آورده به طوری که در معادله تقریبی $Ax \approx b$ صدق کند با این وجود که $A \in \mathbf{R}^{m \times n}$ منفرد بوده یا مسئله بد وضع باشد. در این وضعیت بردار $\hat{x} = A^{-1}b$ (یا در نمونه های رتبه کامل فرامعین $A^\dagger b$ یا شبه معکوس $A^\dagger(A^T A)A^T$) معمولاً یک تقریب نامعتبر برای x است. علاوه بر این حتی زمانی که بردار \hat{x} یک تقریب قابل قبول برای بردار x با $Ax = b$ باشد، خطای برآورد به صورت زیر است [۴]:

$$\|x - \hat{x}\| \leq \|A^{-1}\| \|A\hat{x} - b\|$$

و در حالت فرامعین نیز خطا به صورت زیر خواهد بود: $\|x - \hat{x}\| \leq \|A^\dagger\| \|A\hat{x} - b\|$ حالت هایی وجود دارد که برازش دقیق داده ها ممکن نیست یا حتی داده ها برای به دست آوردن ماتریس $X^T X$ که معکوس پذیر باشد، کافی نیستند یا ممکن است خطایی در داده ها باشد که تلاش برای رسیدن به خروجی بی فایده باشد.

در مسائل بد طرح بعضی پارامترها با روش کمترین مربعات به خوبی تعیین نمی شوند. به ویژه زمانی که تعداد پارامترها بیشتر از کمیت های معلوم باشد، بنابراین در این وضعیت روش کمترین مربعات استاندارد کارآمد نیست، در این حالت ها روشی که غالباً به کار گرفته می شود منظم سازی (پایدار سازی) است. روش منظم سازی برای به دست آوردن برآورد معنا داری از جواب برای مسائل بد طرح به کار گرفته می شود.

منظم سازی در ریاضیات، آمار و به ویژه در شاخه هایی از یادگیری ماشین و مسائل معکوس برای فرآیند به کارگیری اطلاعات اضافی به منظور حل مسائل بد طرح یا به منظور جلوگیری از بیش برازش، مطرح می شود. روش منظم سازی تیخونوف به افتخار آندری تیخونوف^۱ نام گذاری شده است و یک روش رایج به کارگیری منظم سازی برای مسائل بد طرح است. این روش همچنین با نام های رگرسیون ریج^۲ و تیخونوف-میلر یا روش معکوس خطی مقید نیز عنوان می شود.

۱-۲-۲ رگرسیون ریج

رگرسیون ریج متناظر با حل مسئله بهینه سازی زیر است: (می توان با یک ضریب وزنی تاثیر تابع خطا و نرم β را متفاوت در نظر گرفت.)

$$\arg \min_{\beta} L_{\lambda}(\beta) = \lambda \|\beta\|^2 + \sum_{i=1}^l (y_i - g(x_i))^2 \quad (5-1)$$

^۵Ill-posed

^۱Andrey Tikhonov

^۲Ridge regression

یا

$$\arg \min_{\beta} L_{\lambda}(\beta) = \lambda \|\beta\|^2 + \|Y - X\beta\|^2 \quad (6-1)$$

λ یک عدد مثبت است که میزان اهمیت^۱ بین نرم و تابع خطا را مشخص می‌کند و بنابراین درجه‌ی منظم‌سازی را نیز کنترل می‌کند.

لذا مسئله یادگیری به حل یک مسئله بهینه‌سازی در \mathbb{R}^n تبدیل می‌شود.

برای حل مسئله بهینه‌سازی؟؟؟ دو روش وجود دارد: روش اول برآوردی از β را براساس نمایش اصلی رگرسیون ریبج و روش دوم برآوردی از β را مبتنی بر نمایش دوگان ارائه می‌دهد، که در قسمت بعد این دو روش را توضیح خواهیم داد.

۱-۲-۲-۱ نمایش اصلی رگرسیون ریبج

به صورتی مشابه با روش بدست آوردن β در رگرسیون خطی با مشتق‌گیری از تابع هدف (؟؟) نسبت به پارامتر β و برابر صفر قرار دادن آن داریم:

$$\frac{\partial L_{\lambda}(\beta)}{\partial \beta} = 2\lambda\beta - 2X^T Y + 2X^T X\beta = 0$$

$$X^T X\beta + \lambda\beta = (X^T X + \lambda I_n)\beta = X^T Y \quad (7-1)$$

که I_n ماتریس همانی $n \times n$ است. در این حالت ماتریس $(X^T X + \lambda I_n)$ معکوس پذیر است اگر $\lambda > 0$ باشد، بنابراین جواب به صورت زیر حاصل می‌شود:

$$\beta = (X^T X + \lambda I_n)^{-1} X^T Y \quad (8-1)$$

در نتیجه تابع پیش‌بینی^۲ به صورت زیر است:

$$g(x) = \langle \beta, x \rangle = \beta^T x = Y^T X (X^T X + \lambda I_n)^{-1} x$$

^۱trade off

^۲Prediction function

۲-۲-۲-۱ نمایش دوگان رگرسیون ریج

مجددا رابطه؟؟ را در نظر می‌گیریم:

$$X^T X \beta + \lambda \beta = (X^T X + \lambda I_n) \beta = X^T Y$$

با بازنویسی آن بر حسب β داریم:

$$\beta = \lambda^{-1} X^T (Y - X \beta)$$

که اگر α را به صورت زیر در نظر بگیریم:

$$\alpha = \lambda^{-1} (Y - X \beta)$$

خواهیم داشت:

$$\beta = X^T \alpha$$

در ادامه خواهیم دید که می‌توان $g(x)$ را بر حسب ضرب داخلی x و دادگان آموزشی به دست آورد:

$$\alpha = \lambda^{-1} (Y - X \beta)$$

$$\Rightarrow \lambda \alpha = (Y - X X^T \alpha)$$

$$\Rightarrow (X X^T + \lambda I_l) \alpha = Y$$

$$\Rightarrow \alpha = (G + \lambda I_l)^{-1} Y \quad (9-1)$$

که $G = X X^T$ یا به صورت مولفه ای $G_{ij} = \langle x_i, x_j \rangle$ است.

بنابراین در نمایش دوگان

$$\beta = X^T (G + \lambda I_l)^{-1} Y \quad (10-1)$$

است، با فرض این که $k_i = \langle x_i, x \rangle$ باشد در نتیجه تابع پیش بینی به صورت زیر حاصل می شود:

$$g(x) = \langle \beta, x \rangle = \langle \sum_{i=1}^l \alpha_i x_i, x \rangle = \sum_{i=1}^l \alpha_i \langle x_i, x \rangle$$

$$\sum_{i=1}^l \alpha_i k_i = \langle \alpha, K \rangle = \alpha^T K = Y^T (G + \lambda I_l)^{-1} K$$

بنابراین دوروش متمایز برای حل مسئله بهینه سازی رگرسیون ریح از معادله α ؟ ارائه شد. در روش اول معادله α ؟ بردار وزن را محاسبه می کند و به عنوان جواب اصلی شناخته می شود. در حالی که معادله α ؟ جواب را به صورت ترکیب خطی از داده های آموزشی ارائه می دهد و به عنوان جواب دوگان شناخته می شود. پارامتر α ، به عنوان متغیر دوگان معرفی شد. نتیجه واضح از راه حل دوگان رابطه α ؟ آن است که مثال های آموزشی توسط حاصل ضرب داخلی بین جفت نقاط آموزشی در ماتریس $G = X^T X$ مورد استفاده قرار می گیرد.

ماتریس G به عنوان ماتریس گرام^۱ معرفی می شود. ماتریس گرام و ماتریس $(G + \lambda I_l)$ با بعد $l \times l$ می باشند. اگر بعد فضای ویژگی (n) بزرگتر از تعداد نمونه های آموزشی (l) باشد، حل معادله α ؟ از معادله اصلی α ؟ که شامل ماتریس $(X^T X + \lambda I_n)$ از بعد $n \times n$ است، کارآمدتر است.

۳-۱ نداشت غیرخطی

روش رگرسیون ریح که در قسمت قبل مطرح شد، مسئله α مشخص کردن یک رابطه α خطی بین یک متغیر انتخابی و سایر ویژگی ها است، که فرض می شود چنین رابطه ای عملی باشد. با این حال در بسیاری از موارد، روابط مشاهده شده غیر خطی هستند لذا متغیر انتخاب شده فقط بر اساس تابعی غیر خطی از سایر ویژگی ها می تواند به دقت برآورد شود. استراتژی کلی این است سایر ویژگی ها از داده ها را به یک فضای ویژگی جدید تصویر می کنند به طوری که رابطه α تعیین شده، بتواند به صورت خطی ارائه شود. بنابراین الگوریتم رگرسیون ریح توصیف شده در قبل قادر به کشف و آشکارسازی آن ها خواهد بود. نداشت زیر را در نظر بگیرید:

$$\phi : \mathbf{R}^n \longrightarrow F \subseteq \mathbf{R}^N$$

$$x \longmapsto \phi(x)$$

انتخاب نداشت ϕ به منظور تبدیل یک رابطه α غیر خطی به یک رابطه α خطی است. این نداشت انتظار ما از رابطه α $y = \hat{S} = \{(\phi(x_1), y_1), \dots, (\phi(x_l), y_l)\}$ به صورت S ، نتیجه نداشت ϕ تبدیل مجموعه داده های S ، است.

^۱Gram matrix

است.

حال مانند قبل هدف یافتن رابطه ای به فرم زیر است:

$$f((x, y)) = |y - g(x)| = |y - \langle \beta, \phi(x) \rangle| = |\xi|$$

اگرچه روش اصلی می تواند به کار گرفته شود اما مشکلی که به وجود خواهد آمد این است: اگر N خیلی بزرگ باشد؛ پیدا کردن جواب یک سیستم $N \times N$ از معادلات هزینه بر است. از طرفی دیگر اگر جواب های دوگان در نظر گرفته شود؛ نشان داده شد که همه اطلاعات مورد نیاز الگوریتم، حاصل ضرب داخلی بین نقاط داده $\langle \phi(x), \phi(z) \rangle$ در فضای ویژگی F است. به ویژه تابع پیش بینی $g(x) = Y^T(G + \lambda I)^{-1}K$ شامل ماتریس گرام $G + XX^T$ با درایه های

$$G_{ij} = \langle \phi(x_i), \phi(x_j) \rangle \quad (11-1)$$

است. که سطرهای X بردارهای ویژگی $\phi(x_1)^T, \dots, \phi(x_l)^T$ می باشد. بردار K شامل مقادیر

$$k_i = \langle \phi(x_i), \phi(x) \rangle \quad (12-1)$$

است.

گاهی اوقات حاصل ضرب داخلی می تواند به طور موثرتری به عنوان یک تابع مستقیم از ویژگی های ورودی، بدون محاسبه صریح نگاشت ϕ محاسبه می شود. تابعی که این محاسبات مستقیم را انجام می دهد به عنوان تابع کرنل معرفی می شود.

تعریف ۸. (تابع کرنل^۱). κ یک تابع کرنل است که برای همه $x, z \in X$ رابطه زیر برقرار باشد:

$$\kappa(x, z) = \langle \phi(x), \phi(z) \rangle$$

مثال ۱. فضای دو بعدی ورودی $x \subseteq \mathbb{R}^2$ را با نگاشت ویژگی زیر در نظر می گیریم:

$$\phi : x \mapsto \phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \in F = \mathbf{R}^3$$

فضای فرضی تابع خطی F به صورت زیر خواهد بود:

$$g(x) = \omega_{11}x_1^2 + \omega_{22}x_2^2 + \omega_{12}\sqrt{2}x_1x_2$$

^۱Kernel function

نگاشت ویزگی، داده‌ها را از فضای دو بعدی به فضای سه بعدی به گونه‌ای نگاشت می‌کند که روابط خطی در فضای ویزگی متناظر با روابط درجه دو در فضای ورودی هستند. ترکیب نگاشت طرح با حاصل ضرب داخلی در فضای طرح می‌تواند به صورت زیر محاسبه شود:

$$\begin{aligned} \langle \phi(x), \phi(z) \rangle &= \langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) \rangle \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 = \langle x, z \rangle^2 \end{aligned}$$

بنابراین تابع

$$k(x, z) = \langle x, z \rangle^2$$

یک تابع کرنل با فضای ویزگی F است. این به این معنا است که می‌توان حاصل ضرب داخلی بین تصویر دو نقطه در فضای ویزگی را بدون محاسبه صریح مختصات آن‌ها، محاسبه کرد.

اکنون که با روش رگرسیون، رگرسیون ریج، نگاشت غیر خطی و تابع کرنل آشنا شدید به ادامه بحث درهم تنیدگی می‌پردازیم. قابل ذکر است که درهم تنیدگی به ادامه مسائل ذکر شده می‌پردازد.

۴-۱ کارهای پیشین

روش‌های متفاوتی در زمینه درهم تنیدگی تصاویر ارائه شده است. در این بخش به توضیح تعدادی از این روش‌ها می‌پردازیم.

۱-۴-۱ درهم تنیدگی دودویی

همانطور که در قبل ذکر شده است بی‌نهایت جواب برای مسئله درهم تنیدگی وجود دارد. برای اینکه محدوده جواب را کمتر کنیم تعدادی محدودیت در نظر می‌گیریم. α می‌تواند مقداری بین ۰ و ۱ را داشته باشد. اگر α مقدار ۰ را به خود بگیرد پیکسل به طور قطع پیش زمینه یا ۱ داشته باشد، پس زمینه است که به این نوع تقسیم بندی، درهم تنیدگی دودویی^۱ می‌گویند [۴].

محدودیت دیگر این است که کاربر یک تقسیم بندی اولیه را انجام دهد و فضای پیش زمینه و پس زمینه را مشخص کند. واضح است در صورت نبود هیچگونه محدودیتی، بی‌نهایت راه حل برای این کار وجود دارد. مسئله درهم تنیدگی، برآورد F, B, α در معادله؟؟ در مناطق بدون برچسب است. نتیجه عمل درهم تنیدگی در تصویر؟؟ نشان داده شده است.

^۱binary matting



(ج) خروجی

(ب) نقشه سه گانه

(آ) تصویر اصلی

شکل ۱-۳: نمونه عمل درهم‌تنیدگی تصویر

اگرچه درهم‌تنیدگی یک مسئله بدوضع است ولی همبستگی قوی بین پیکسل‌های نزدیک به هم می‌تواند اهرمی برای کاهش مشکلات باشد. به طور معمول فرض بر این است، پیکسل‌هایی که دارای رنگ‌های مشابه هستند اغلب پارامترهای درهم‌تنیدگی مشابهی دارند (مانند α). این همبستگی در بسیاری از کاربردها استفاده شده است. مانند؛ حذف نویز، تفکیک پذیری، رنگ‌بندی، تقسیم‌بندی و ... [۴].

هنگامی که دو پیکسل به هم نزدیک باشند قطعا F و B و I آن‌ها به هم نزدیک است پس از یک پیکسل شناخته شده می‌توان میزان α یک پیکسل ناشناخته را با استفاده از روش‌هایی من جمله؛ روش فرم بسته، روش پواسن، روش می‌شیمما و... تخمین زد.

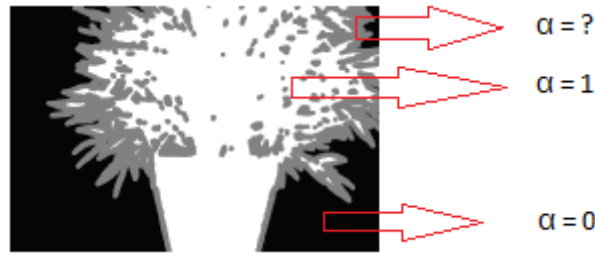
در هر روش در ابتدا سوالاتی که باید به آنها پاسخ داد به این شرح است: چگونه پیکسل همسایه را بیابیم؟ به عبارتی؛ در چه فاصله‌ای نمونه پیش‌زمینه و پس‌زمینه به درستی قابل شناسایی است؟ به چه صورت نمونه‌ها جمع‌آوری می‌شوند؟ چگونه می‌توانیم F و B مربوط به هر پیکسل رابه درستی برآورد کنیم؟ روش‌های متفاوتی برای مقابله با این مشکلات وجود دارد که در بخش‌های آتی شرح داده خواهند شد. با توجه به پیکسل شناخته شده این مدل‌ها فرض را بر این می‌گیرند که فاصله پیش‌زمینه و پس‌زمینه از یک توزیع آماری تبعیت می‌کنند.

۲-۴-۱ روش درهم‌تنیدگی مبتنی بر یادگیری

درهم‌تنیدگی شامل ۲ وظیفه مهم است: برآورد آلفا و محاسبه رنگ پیش‌زمینه و پس‌زمینه می‌باشد. فرض کنید تصویر I داده شده است، مجموعه همه پیکسل‌هایش را با $\Omega = \{1, \dots, n\}$ نشان داده که n تعداد کل پیکسل‌ها است. مجموعه پیکسل‌های برجسب‌گذاری شده با $\Omega_l \subset \Omega$ نمایش می‌دهیم، که مقدار α برای آن‌ها معلوم است. برآورد آلفا به عنوان محاسبه مقدار α برای مجموعه‌ای از پیکسل‌های نامعلوم (برجسب‌گذاری نشده) $\Omega_u = \Omega - \Omega_l$ تعریف می‌شود، که در آن مجموعه همه پیکسل‌های برجسب‌گذاری شده، مرکب از ۲ زیرمجموعه است:

۱. Ω_l^F برجسب‌گذاری شده به عنوان قطعا پیش‌زمینه که برای آن‌ها $\alpha = 1$ است.

۲. Ω_l^B برجسب‌گذاری شده به عنوان قطعا پس‌زمینه که برای آن‌ها $\alpha = 0$ است.



شکل ۱-۴: نقشه سه گانه

که در تصویر؟؟ Ω_l^B و Ω_l^F به ترتیب با رنگ های سفید و سیاه نشان داده شده است. هر پیکسل $i \in \Omega$ به عنوان یک نقطه داده تعبیر می شود که با $x_i \in R^d$ نمایش داده می شود. x_i می تواند همان I_i یا سایر ویژگی های به دست آمده برای پیکسل i باشد.

در این بخش $x_i = I_i$ در نظر گرفته می شود، که همان شدت رنگ خاکستری پیکسل یا یک بردار مرکب از مولفه رنگ برای تصاویر رنگی است.

مسئله یادگیری می تواند به صورت زیر فرمول بندی شود:

مجموعه نقاط $\chi \subseteq R^d$ که $\chi = \{x_i\}_{i \in \Omega}$ مفروض اند و $\{\alpha_i\}_{i \in \Omega_l}$ مقادیر آلفا برای داده های برجسب گذاری شده $\chi_l = \{x_i\}_{i \in \Omega_l}$ می باشد. هدف، محاسبه دقیق مقدار آلفا $\{\alpha_i\}_{i \in \Omega_u}$ برای داده های برجسب گذاری نشده، $\Omega_u = \Omega - \Omega_l$ ، از روش یادگیری است.

روش درهم تنیدگی بر اساس یادگیری محلی یک مدل آلفا-رنگ را برای هر پیکسل در تصویر فقط بر اساس نزدیکترین پیکسل ها که تصور می شود بیش ترین ارتباط را دارند، ارائه می دهد [؟]. سپس برآورد آلفا با این فرض که مقدار آلفای هر پیکسل ترکیب خطی از مقدار آلفای پیکسل های مجاورش است، فرمول بندی می شود. ادامه این روش درهم تنیدگی می تواند با درهم تنیدگی محلی مانند استفاده از رگرسیون خطی، رگرسیون ریج، تابع کرنل و یا درهم تنیدگی غیر محلی انجام شود و مقدار آلفا محاسبه شود.

۳-۴-۱ الگوریتم روزان و توماسی

یک الگوریتم درهم تنیدگی توسط روزان^۱ و توماسی^۲ در سال ۲۰۰۰ پیشنهاد شد [؟]. اگر مسیر مرزهای بین توزیع رنگ هر یک از اشیاء را چند برابر کنیم، مقدار آلفا در این مسیر اندازه گیری می شود. همانطور که در شکل؟؟ نشان داده شده است، این روش به طور ضمنی فرض می کند که ناحیه نامشخص یک نوار باریک دور ناحیه پیش زمینه است و اسکلت منطقه ناشناخته را می توان با یک زنجیره ای از پیکسل ها نشان داد.

ساخت مدل و روش برآورد آلفا در الگوریتم؟؟ شرح داده شده است. طبق الگوریتم ذکر شده و معادله؟؟ :

$$I = \alpha F - \alpha B + B$$

^۱Ruzon

^۲Tomasi

$$I - B = \alpha(F - B)$$

در نتیجه:

$$\alpha = \frac{I - B}{F - B} \quad (13-1)$$

از طریق معادله فوق می توان میزان آلفا را محاسبه کرد. از شرح الگوریتم، روشن است که تعدادی از فرضیه های ضعیف در

الگوریتم ۱-۱ الگوریتم ساخت مدل و برآورد آلفا به روش روزان و توماسی .

ورودی: تصویر اصلی،

خروجی: مقدار آلفا.

- ۱: زنجیره ای از پیکسل ها (اسکلت منطقه ناشناخته) را به فواصل با نقاط پایه انتخاب شده تقسیم کنید.
- ۲: بر روی هر نقطه پایه متمرکز شده، یک پنجره مکانی محلی که منطقه ناشناخته محلی را پوشش می دهد، تعریف می شود و منطقه پیش زمینه و پس زمینه محلی را پوشش می دهد.
- ۳: پیکسل های پیش زمینه و پس زمینه در پنجره محلی با توزیع گاوسی تخمین زده می شوند.
- ۴: سپس با پیش زمینه و پس زمینه هایی که با توزیع گاوسی تخمین زده شده اند یک مجموعه بسازید. این در حالی است که برخی از ارتباطات با توجه به معیارهای خاص "تقاطع" و "زاویه" رد می شوند، این خاصیت در خطوط بین توده های نقطه در شکل؟؟ نشان داده شده است.
- ۵: رنگ مشاهده شده از یک پیکسل ناشناخته به عنوان یک توزیع متوسط بین توزیع پیش زمینه و پس زمینه مدلسازی می شود. توزیع متوسط نیز به عنوان مجموع گاوسین ها تعریف می شود، طوری که در هر گاوسین به طور خطی بر اساس مقدار آلفا تخمین زده شده، میانگین و کوواریانس بین جفت گاوسین پیش زمینه و پس زمینه درونیابی می شود. این مسئله در شکل؟؟ نشان داده شده است.
- ۶: آلفای بهینه آن است که متوسط مقدار آلفای همسایگانش باشد و رنگی را که شناسایی می کند دارای حداکثر احتمال باشد.
- ۷: برای یک پیکسل ناشناخته، بعد از آنکه مقدار آلفای آن تخمین زده می شود، رنگ پیش زمینه آن با درونیابی جفت گاوسین پیش زمینه و پس زمینه برآورد می شود. این برآورد به این صورت است که F و B برای پیکسل ناشناخته با توجه به گاوسین F و B شناخته شده به دست می آید.

این رویکرد به کار برده شده است. به عنوان مثال، مقادیر آلفا به طور مستقل برای پیکسل های ناشناخته محاسبه می شود که می تواند منجر به اختلالات در محاسبه آلفا نهایی شود.

۴-۴-۱ درهم تیدگی مبتنی بر نظریه بیز

بر اساس الگوریتم روزان و توماسی، چوانگ و همکارانش پیشنهاد یک رویکرد درهم تیدگی مبتنی بر نظریه بیز^۱ در سال

۲۰۰۱ دادند.

^۱Bayesian Matting

مشابه الگوریتم روزان و توماسی، این رویکرد با اندکی بهبود، رنگ پیش زمینه و پس زمینه را به عنوان مخلوط گاوسی مدلسازی می‌کند. ابتدا، همانطور که در شکل؟؟ نشان داده شده است، درهم تنیدگی مبتنی بر نظریه بیز از یک پنجره برای تعریف همسایگی استفاده می‌کند. علاوه بر استفاده از نمونه پیش زمینه و پس زمینه برای ساخت توزیع رنگ، از F_z و B_z و α_z محاسبه شده نزدیک به پیکسل جاری نیز استفاده می‌شود به طوریکه هر پیکسل در هر همسایگی، در محاسبه گاوسین پیش زمینه و پس زمینه مشارکت خواهد کرد. گذشته از این، این نوع از مسئله درهم تنیدگی در یک چارچوب بیزی به خوبی تعریف شده، فرموله شده است و این رویکرد با استفاده از روش پیشنهادی پیشنهادی حل شده است.

به صورت ریاضی برای هر پیکسل ناشناخته مقادیر F_z و B_z و α_z و I_z به صورت زیر محاسبه می‌شود [۱۴].

$$\begin{aligned} & \arg \max_{F_z, B_z, \alpha_z} P(F_z, B_z, \alpha_z | I_z) \\ & = \arg \max_{F_z, B_z, \alpha_z} L(I_z | F_z, B_z, \alpha_z) + L(F_z) + L(B_z) + L(\alpha_z) \end{aligned} \quad (14-1)$$

$L(\cdot) = \log P(\cdot)$ است. اولین قسمت به صورت $L(I_z | F_z, B_z, \alpha_z) = ||I_z - \alpha_z F_z - (1 - \alpha_z) B_z||^2 / \sigma_z^2$ محاسبه می‌شود این در حالی است که واریانس رنگ σ_z بصورت محلی محاسبه می‌شود. برای برآورد $L(F_z)$ ، رنگ‌های پیش زمینه در ناحیه ی مجاور، ابتدا به صورت گروهی تقسیم بندی می‌شوند و در هر گروه یک گاوس بیضی جهت دار^۱ با محاسبه میانگین \bar{F} و کوواریانس \sum_F تخمین زده می‌شود.

$L(F_z)$ به صورت $-(F_z - \bar{F})^T \sum_F^{-1} (F_z - \bar{F}) / 2$ تعریف می‌شود. $L(B_z)$ نیز به همان صورت با استفاده از نمونه پس زمینه محاسبه می‌شود.

$L(\alpha)$ یک ثابت است. در ابتدا میزان α یک ثابت دلخواه در نظر گرفته می‌شود. معادله با استفاده از گام های زیر و تکرار برآورد F_z و B_z محاسبه می‌شود:

۱. محاسبه F_z و B_z براساس α_z

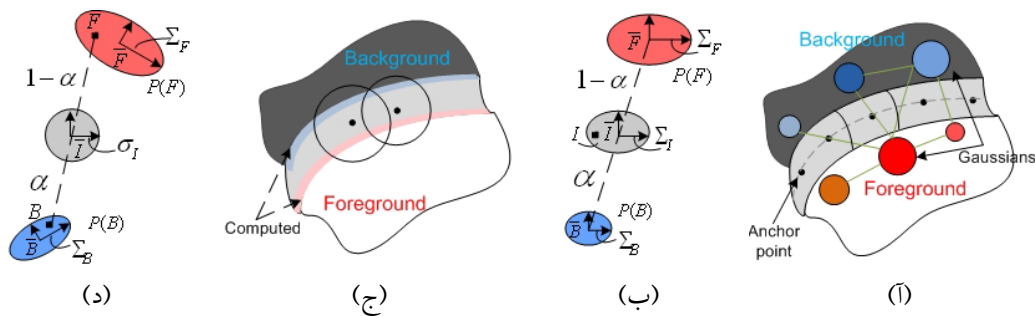
$$\begin{bmatrix} \sum_F^{-1} + I\alpha_z^2 / \sigma_z^2 & I\alpha_z(1 - \alpha_z) / \sigma_z^2 \\ I\alpha_z(1 - \alpha_z) / \sigma_z^2 & \sum_B^{-1} + I(1 - \alpha_z)^2 / \sigma_z^2 \end{bmatrix} \begin{bmatrix} F \\ B \end{bmatrix} = \begin{bmatrix} \sum_F^{-1} \bar{F} + I\alpha_z / \sigma_z^2 \\ \sum_B^{-1} \bar{B} + I(1 - \alpha_z) / \sigma_z^2 \end{bmatrix} \quad (15-1)$$

۲. محاسبه α_z براساس F_z و B_z

$$\alpha_z = \frac{(I_z - B_z)(F_z - B_z)}{||F - B||^2} \quad (16-1)$$

از لحاظ خوشه بندی چندگانه پیش زمینه و پس زمینه، این فرآیند برای هر جفت خوشه پیش زمینه و پس زمینه انجام می‌شود که ملاک انتخاب جفت، حداکثر میزان شباهت می‌باشد. زمانیکه فرض ها بر اساس یک تصویر ورودی و یک نقشه سه

^۱oriented elliptical Gaussian



شکل ۱-۵:؟؟ تصویر روشن روزان و توماسی.؟؟ مدل درونی روشن روزان و توماسی.؟؟ محدوده تعیین شده Bayesian matting.؟؟ برآورد MAP پارامترهای درهم تنیدگی در Bayesian matting.

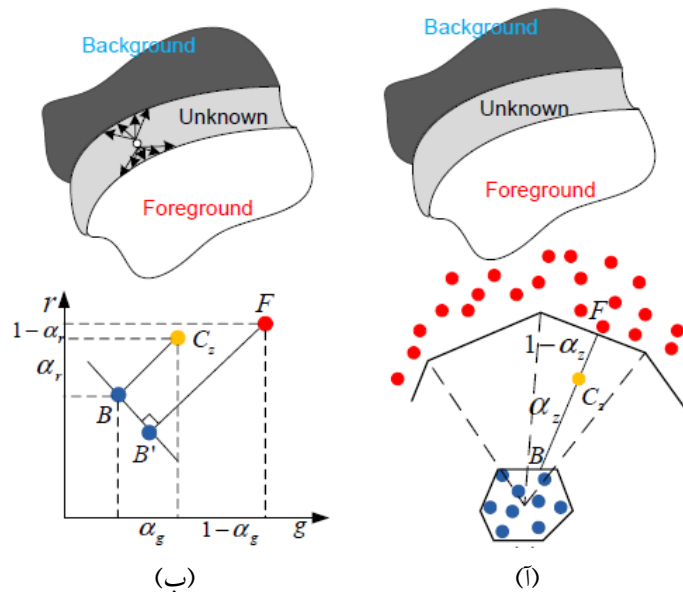
گانه‌ی به خوبی مشخص شده باشد الگوریتم بیزی می تواند خروجی دقیقی را تولید کند. به عنوان مثال، در مواردی مانند اینکه تصویر ورودی حاوی مناطق دارای بافت ناهموار است که استفاده از گاوسین برای مدل سازی ارقام بالای توزیع رنگ مناسب نیست، یا استفاده از نقشه سه گانه درشت به طوری که ارتباطات بین پیکسل های ناشناخته و پیش زمینه و پس زمینه نمونه ها ضعیف هستند، نتایج بسیار پر خطا تولید می شود. در جدول؟؟ از نقشه سه گانه درشت و ریز جهت تنیدگی استفاده شده است که می توان میزان خطا را مشاهده کرد. مشاهده می شود که استفاده از نقشه سه گانه درشت خطای بیشتری نسبت به نقشه سه گانه ریز دارد.

تصویر ورودی	نقشه سه گانه درشت	نقشه سه گانه ریز
نتیجه مصور		
میزان خطا	۱۱۹۸.۹۶۴۸	۸۴۲.۵۶۸۵

جدول ۱-۱: نتایج سه روش درهم تنیدگی

۵-۴-۱ روش میشیما

میشیما [؟] تکنیک درهم تنیدگی بر روی یک صفحه نمایش آبی را براساس نمونه های پیش زمینه و پس زمینه توسعه داد. همانطور که در شکل؟؟ نشان داده شده است، با استفاده از نقشه سه گانه از آنجایی که پس زمینه تنها یک خوشه رنگ دارد، تمام پیکسل های پس زمینه را می توان با یک فضای کوچک تقریبی، با یک چندوجهی در فضای رنگ، پوشش داد. تمامی پیکسل های پیش زمینه نیز یک چندوجهی را خارج از پس زمینه تشکیل می دهند. سپس، مقدار آلفا در یک پیکسل ناشناخته توسط محاسبه‌ی موقعیت نسبی دو چندوجهی پس زمینه و پیش زمینه، تخمین زده می شود.



شکل ۱-۶: تصاویر روش های نمونه گیری غیر پارامتری؟؟ می‌شیم و؟؟ حذفی

۶-۴-۱ روش حذفی

سیستم حذفی [؟] عمل برونیابی رنگ های پیش زمینه و پس زمینه، در محدوده ناشناخته را به صورت زیر انجام می دهد. برای هر پیکسل ناشناخته I ، رنگ پیش زمینه F_z به عنوان یک مجموع وزنی از رنگ های پیش زمینه شناخته شده موجود محاسبه می شود که این وزن ها متناسب با فاصله فضایی پیکسل ها با I است. رنگ پس زمینه B_z بار اول به عنوان B'_z محاسبه می شود. پس از آن با توجه به موقعیت نسبی I و F_z ، همانطور که در شکل؟؟ نشان داده شده است، بازیابی می شوند. سپس مقدار α سه بار و هر بار در یک کانال رنگی محاسبه می شود. به عنوان مثال در کانال قرمز آلفا به صورت زیر برآورد می شود:

$$(r(I_z) - r(B_z)) / (r(F_z) - r(B_z))$$

که در آن $r(\cdot)$ مقدار کانال قرمز رنگ است.

در نهایت α_z نهایی به عنوان یک مجموعه وزن دار از این مقادیر برآورد می شود که این وزن متناسب با اختلاف پیش زمینه و پس زمینه در کانال رنگ مربوطه است.

۷-۴-۱ روش پواسن

در مدل پواسن [؟]، گرادیان درهم تنیدگی با این فرض مدلسازی می شود که تغییرات شدت روشنایی در پیش زمینه و پس زمینه به صورت محلی، هموار است.

با در نظر گرفتن مشتقات جزئی در دو طرف معادله؟؟ به دست می آید:

$$\nabla I_z = (F_z - B_z)\nabla\alpha_z + \alpha_z\nabla F_z + (1 - \alpha_z)\nabla B_z \quad (17-1)$$

که در آن $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ عملگر گرادیان است.

از آنجایی که F_z و B_z هموار فرض می شوند و $\alpha_z\nabla F_z + (1 - \alpha_z)\nabla B_z$ در مقایسه با $(F_z - B_z)\nabla\alpha_z$ نسبتاً کوچک هستند، گرادیان درهم تنیدگی می تواند به صورت زیر تقریب زده شود:

$$\nabla\alpha_z = \frac{1}{(F_z - B_z)}\nabla I_z \quad (18-1)$$

به منظور برآورد گرادیان ابتدا نیاز است $F_z - B_z$ برآورد شود.

F_z و B_z به عنوان رنگ پیش زمینه و پس زمینه برای پیکسل ناشناخته انتخاب می شوند. با حل معادلات پواسن در شبکه تصویر، نتیجه نهایی به صورت زیر است:

$$\alpha^* = \arg \min_{\alpha} \iint_{z \in \Omega} \left\| \nabla\alpha_z - \frac{1}{F_z - B_z} \nabla I_z \right\|^2 dz \quad (19-1)$$

Ω محدوده نامشخص در نقشه سه گانه است. یکی از محدودیت های اصلی سیستم پیشنهادی، روش تخمین $F_z - B_z$ است.

۸-۴-۱ روش فرم بسته

روش فرم بسته [؟] محدودیت مسئله پواسن که تخمین مقدار $F_z - B_z$ است را با حذف F و B حل کرده است. این روش یک تابع هزینه درجه دوم را در α ارائه می دهد که می تواند به آسانی به عنوان یک سیستم خطی معادلات حل شود. فرض اساسی این رویکرد این است که هر F و B یک ترکیب خطی از دو رنگ در یک پنجره کوچک (معمولاً 3×3 یا 5×5) در اطراف هر پیکسل است که به عنوان مدل خطی رنگ اشاره شده است. نشان داده شده است که تحت این فرض، مقادیر آلفای در یک پنجره کوچک w می تواند به صورت زیر بیان شود.

$$\alpha_i = \sum_c a^c I_i^c + b, \forall i \in \omega \quad (20-1)$$

که c به کانال های رنگ اشاره دارد، و a^c و b ثابت در پنجره هستند. سپس تابع هزینه y درهم تنیدگی به صورت زیر تعریف می شود.

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in \omega_j} \left(\alpha_i - \sum_c a_j^c I_i^c - b_j \right)^2 + \epsilon \sum_c a_j^c \right) \quad (21-1)$$

علاوه بر این، a^c و b را می توان از تابع هزینه حذف کرد، و تنها یک مقدار درجه دوم را از α به ارمغان می آورد:

$$J(\alpha) = \alpha^T L \alpha \quad (22-1)$$

که L یک ماتریس $N \times N$ است که عنصر (i, j) آن:

$$\sum_{k | (i,j) \in \omega_k} \left(\delta_{ij} - \frac{1}{|\omega_k|} \left(1 + (I_i - \mu_k) \left(\sum_k + \frac{\epsilon}{|\omega_k|} I_3 \right)^{-1} (I_j - \mu_k) \right) \right) \quad (23-1)$$

که \sum_k یک ماتریس کوواریانس 3×3 است، μ_k یک بردار میانگین رنگ 1×3 در یک پنجره w_k است و I_3 ماتریس اصلی 3×3 است. ماتریس L ، که به نام لاپلاسیان درهم تنیدگی نامیده می شود، مهمترین نتیجه تحلیلی این رویکرد است. فرم بسته اگر بخواهد چند لایه را جداسازی کند آن گاه تابع هزینه بر حسب آلفا به صورت زیر محاسبه می شوند:

$$g(\alpha) = \operatorname{argmin} \alpha^T L \alpha + \lambda (\alpha^T - m^T) D (\alpha - m) \quad , \quad s.t. \alpha_i = 1 \quad \text{or} \quad 0, \quad \forall i \in \partial \Omega \quad (24-1)$$

که m یک بردار دودویی از همه شاخص های پیکسل های تصویر است. یعنی شامل پیکسل های نشانه گذاری شده و مابقی پیکسل ها است. D یک ماتریس قطری است که عناصر مورب آن اگر 1 باشند برای پیکسل های نشانه گذاری شده و اگر صفر باشند برای مابقی پیکسل ها است. از رابطه؟؟ می توان به صورت زیر نتیجه گرفت.

$$(L + \lambda D) \alpha = \lambda m \quad (25-1)$$

در نهایت از رابطه؟؟ می توان مقدار α را محاسبه کرد.

۹-۴-۱ روش درهم تنیدگی سراسری

روش هایی که تا اکنون ذکر شد از مدل همسایگی محلی استفاده می کنند. با این فرض که محدوده نامشخص یک نوار باریک، در اطراف مرز پیش زمینه است. بنابراین؛ پیکسل های پیش زمینه و پس زمینه در یک پنجره محلی متعلق به هر پیکسل ناشناخته قرار می گیرند.

اگر از نقشه سه گانه بسیار درشت استفاده شود نمونه های پیش زمینه و پس زمینه از نمونه ناشناخته بسیار دور هستند و نمونه ناشناخته قابل تشخیص نمی باشد [؟].

یک رویکرد مدل سراسری، تحت عنوان درهم تنیدگی غیر محلی [؟] است که به صورت غیر محلی عمل درهم تنیدگی را انجام می دهد. درهم تنیدگی غیر محلی سعی بر این دارد که به میزان قابل توجهی میزان دخالت کاربر را در مسئله درهم تنیدگی کم کند. این روش برای حل مشکل تصاویر دارای نویز معرفتی شد که اگر به صورت محلی درهم تنیدگی روی آن ها انجام شود دچار خطای زیادی می شوند زیرا نویز در تصاویر درهم تنیدگی را معطوف به خود می کنند و به اشتباه عمل درهم تنیدگی انجام می شود. این روش به طرز چشمگیری تعداد پیکسل هایی را که کاربر به صورت دستی برچسب گذاری می کند، کاهش می دهد. در این روش از یک روش خوشه بندی پیکسل های تصویر برای برچسب گذاری های کاربر، جهت حذف برچسب گذاری های زائد و غیر ضروری استفاده می شود. اساس کار این روش این است که تصویر را به یکسری قطعات تقسیم بندی می کند و بردار ویژگی را برای هر قطعه به صورت مجزا در نظر می گیرد. این بردار ویژگی به صورت زیر معرفی می شود:

$$\mathbf{X} = \begin{bmatrix} X^R(\tau_1) & X^G(\tau_1) & X^B(\tau_1) & 1 \\ & \vdots & & \\ X^R(\tau_m) & X^G(\tau_m) & X^B(\tau_m) & 1 \end{bmatrix} \quad (26-1)$$

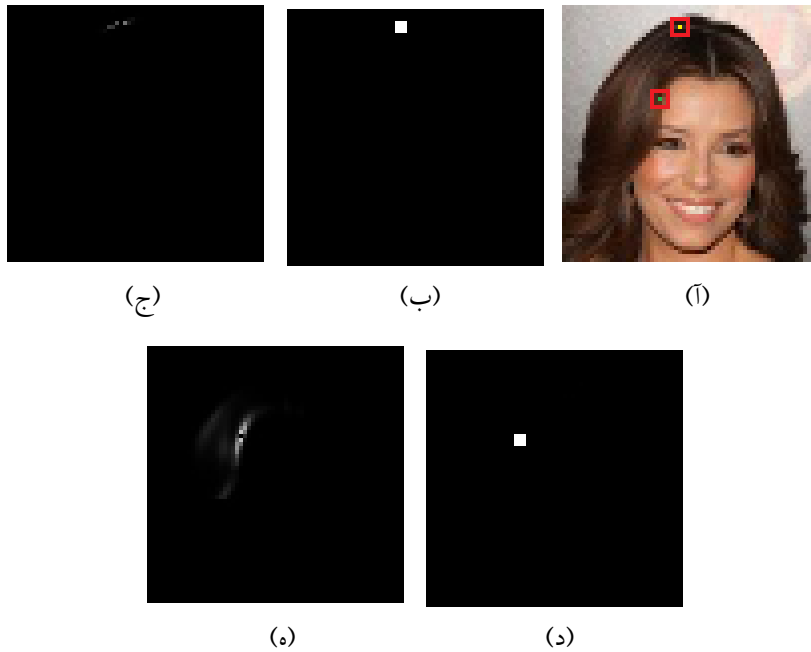
که $\{\tau_1, \dots, \tau_m\}$ یک قطعه از پیکسل های اطراف پیکسل i است در این روش مولفه آلفا برای هر پیکسل به صورت یک مجموعه از آلفاهای یک قطعه اطراف آن پیکسل است:

$$\alpha_i = [\alpha(\tau_1), \dots, \alpha(\tau_m)]^T$$

طبق این درهم تنیدگی، هر پیکسل i یک مجموعه وزن دار از پیکسلهایی است که ظاهری مشابه دارند. این مجموعه وزن دار را به عنوان تابع کرنل معرفی می کنند. در ادامه کار برای تخمین هر X_i از فرمول زیر استفاده می شود:

$$E[X_i] \approx \sum_j X(j)k(i, j) \frac{1}{D_i} \quad (27-1)$$

$$k(i, j) = \exp\left(-\frac{1}{h^2} \|X(i) - X(j)\|^2 - \frac{1}{h^2} d_{ij}^2\right) \quad (28-1)$$



شکل ۱-۷: تصویر اصلی؟؟ همسایگان محلی پیکسل (۵،۲۹)،؟؟ همسایگان غیر محلی پیکسل (۵،۲۹)،؟؟ همسایگان محلی پیکسل (۲۴،۲۳)،؟؟ همسایگان غیر محلی پیکسل (۲۴،۲۳)،؟؟.

$$D_i = \sum_j k(i, j) \quad (29-1)$$

به طوریکه $X(i)$ یک بردار ویژگی محاسبه شده با استفاده از اطلاعات اطراف پیکسل i است، D_{ij} فاصله بین دو پیکسل i, j است و h_1, h_2 ثوابتی تجربی هستند. برای اینکه نحوه عملکرد تابع کرنل را تصور کنید به تصویر؟؟ دقت کنید:

در تصویر؟؟ همسایگان محلی دو پیکسل با همسایگان غیر محلی آن پیکسل ها معرفی شده اند. پیکسل اول (۵،۲۹) که با رنگ زرد و پیکسل دیگر (۲۴،۲۳) که با رنگ سبز مشخص شده است. در همسایگی محلی می توان از چهار همسایگی، که شامل همسایه بالا، همسایه پایین، همسایه چپ و همسایه راست می شود، یا از ۸ همسایگی که شامل همسایگان چهار همسایگی به علاوه همسایه شمال شرقی، همسایه شمال غربی، همسایه جنوب شرقی و همسایه جنوب غربی است، استفاده کرد. در همسایگی غیر محلی تعریف همسایگی به صورت سراسری است یعنی پیکسل های همسایه به میزان شباهت یا نزدیکی شان یک شدت رنگ دارند. هر چقدر همسایگی نزدیکتر باشد شدت رنگ سفید بیشتر است.

به طور کلی روش درهم تنیدگی غیر محلی یک توزیع داده ای گسسته $k(i, \cdot) / D_i$ برای هر پیکسل معرفی می کند و نشان می دهد که هر عکس می تواند به صورت تنک با چند تکه فضایی توصیف شود. این روش از این اصول جهت بدست آوردن میزان آلفا استفاده می کند. همانند تخمین X_i تخمین آلفا را می توان انجام داد. طبق معادله؟؟:

$$E[\alpha_i] \approx \sum_j \alpha_j k(i, j) \frac{1}{D_i} \quad \text{or} \quad D_i \alpha_i \approx k(i, \cdot)^T \alpha \quad (30-1)$$

در نهایت

$$D\alpha \approx A\alpha \quad (31-1)$$

که $A = k(i, j)$ یک ماتریس قطری $N \times N$ و N تعداد پیکسل های تصویر است. بنابراین؛ $(D - A)\alpha \approx 0$ یا $\alpha^T L_c \alpha \approx 0$ که $L_c = (D - A)^T (D - A)$ خوشه بندی لاپلاسیان نامیده می شود.

فصل ۲

جستجوی نزدیکترین همسایگی

یکی از روش های درهم تنیدگی استفاده از نزدیکترین همسایگی است. در این فصل روش نزدیکترین همسایگی و در فصل ۳ روش درهم تنیدگی با نزدیکترین همسایگی شرح داده و در فصل ۴ نتایج این روش با روش های دیگر مقایسه می شود.

۱-۲ نزدیکترین همسایگی

نزدیکترین همسایگی، شامل یافتن نزدیکترین داده های آزمایشی نسبت به داده های آموزشی x در قالب \hat{Y} است. به عبارت دیگر؛ k نزدیکترین همسایگی به صورت زیر تعریف می شود:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \quad (1-2)$$

که $N_k(x)$ همان همسایگان x است که با k نزدیکترین نقاط x_i در نمونه های آموزشی تعریف شده است [۹]. انطباق، مستلزم یک شاخص اندازه گیری است. برای یافتن نزدیکترین ما نیازمند داشتن یک فاصله هستیم که بتوانیم از بین فواصل کمترین فاصله را انتخاب کنیم. مترهای متفاوتی برای اندازه گیری وجود دارد که به طور خلاصه به صورت زیر هستند:

- فاصله اقلیدسی: در ریاضیات، فاصله اقلیدسی، فاصله معمولی بین دو نقطه است. در ادبیات قدیمی تر به عنوان مترادف فاصله فیثاغورث اشاره شده است. طبق این نوع از شاخص اندازه گیری، فاصله بین دو نقطه p, q به صورت زیر محاسبه می شود:

$$d(p, q) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

که در واقع برای یافتن نزدیکترین همسایگی به صورت پیش فرض از این نوع شاخص اندازه‌گیری استفاده می‌شود [۴].

- فاصله Minkowski: یک شاخص اندازه‌گیری در یک فضای برداری نرم شده است که می‌تواند به عنوان تعمیم هر دو فاصله اقلیدسی و فاصله منهن در نظر گرفته شود. این نوع از فاصله به صورت زیر اندازه‌گیری می‌شود:

$$D(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$$

فاصله Minkowski به طور معمول با p بین ۱ و ۲ استفاده می‌شود که $p = ۱$ به فاصله منهن و $p = ۲$ به فاصله اقلیدس مربوط می‌شود.

- فاصله City Block: فاصله City Block بین دو نقطه x, y با K بعد به صورت زیر محاسبه می‌شود:

$$\sum_{j=1}^K |x_j - y_j|$$

این نوع از فاصله همواره بزرگتر و یا برابر صفر است. در مواردی که دو نقطه با هم متشابه باشند فاصله بینشان برابر صفر است. این نوع از فاصله بیشتر به عنوان فاصله منهن^۱ شناخته شده است. همچنین می‌توان گفت نمونه خاصی از فاصله Minkowski با $p = ۱$ است [۴]. اساس کار این نوع از فاصله این است که برای رفتن به دو نقطه در یک شهر که نمی‌توان یک فاصله مستقیم را در نظر گرفت، باید از کنار ساختمان‌ها گذر کرد.

- فاصله چیشف^۲: فاصله چیشف، یک شاخص اندازه‌گیری فاصله است که در آن فاصله بین دو بردار بزرگترین اختلاف آنها در طول هر بعد مختصات است. همچنین به عنوان فاصله شطرنج شناخته شده است. زیرا در بازی شطرنج، حداقل تعداد حرکت‌هایی که پادشاه برای پیروزی از یک مربع در یک صفحه شطرنج به دیگری نیاز دارد، برابر با فاصله چیشف بین مراکز مربع‌ها است [۴].

فاصله چیشف بین دو نقطه یا دو بردار p, q به صورت زیر محاسبه می‌شود:

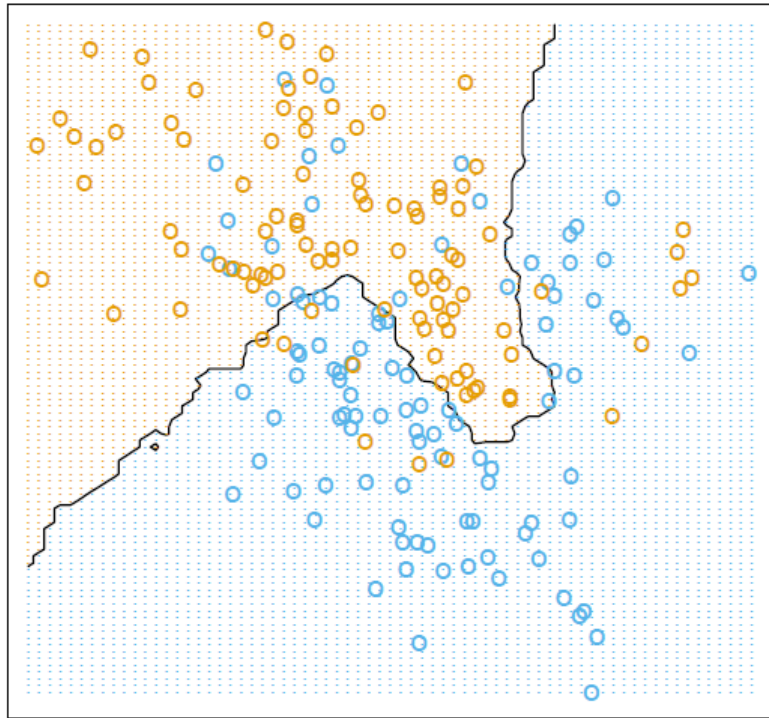
$$D(p, q) := \max_i (|x_i - y_i|)$$

- فاصله همینگ^۳: فاصله همینگ بین دو رشته با طول یکسان، تعداد نمادهای متفاوت در موقعیت یکسان از آن رشته‌ها است. به عبارت دیگر، حداقل تغییرات مورد نیاز برای تغییر یک رشته را به دیگری یا حداقل تعداد خطاها که می‌تواند یک رشته را به دیگری تبدیل کند، اندازه‌گیری می‌کند. به طور مثال: فاصله بین دو رشته ۲۱۷۳۸۹۶ و ۲۲۳۳۷۹۶ برابر ۳ است [۴].

^۱Manhattan Distance

^۲Chebyshev

^۳Hamming Distance



شکل ۲-۱: کلاس طبقه بندی شده با ۱۵ همسایگی

- فاصله Jaccard: یک شاخص اندازه گیری فاصله است که میزان تشابه و یا تمایز بین دو مجموعه داده را بدست می آورد. این شاخص به صورت زیر اندازه گیری می شود [۴]:

$$J(A, B) = \frac{A \cap B}{A \cup B}$$

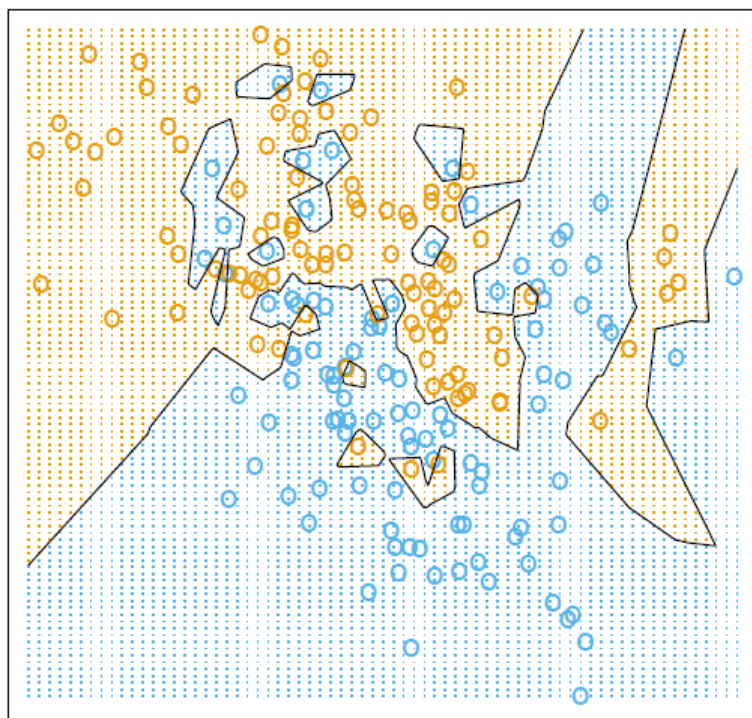
$$d_J(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

برای یافتن نزدیکترین همسایگی، ما k نزدیکترین مشاهدات x_i را نسبت به ورودی x می یابیم و سپس به طور متوسط پاسخ آنها را می سنجم.

در شکل ۲؟ با استفاده از پیاده سازی کد متلب روی یک مجموعه داده، ۱۵ نزدیکترین همسایگی به عنوان روش مناسب معرفی شده است. مناطق رنگی نشان می دهد که همه نقاط با ارزیابی روش در یک شبکه و فضای ورودی خوب به واسطه چنین قواعدی به عنوان آبی یا نارنجی طبقه بندی می شوند. مشاهده می شود که مرز تصمیم گیری که آبی را از مناطق نارنجی جدا می کند بسیار نامنظم است و تنها به نواحی محلی پاسخ می دهد.

شکل ۲؟ نتایجی را برای طبقه بندی ۱ نزدیکترین همسایگی نشان می دهد: اگر نزدیکترین نقطه را xl, yl بنامند، \hat{Y} مقدار yl نزدیکترین نقطه xl به x در داده های آموزشی است. در این حالت، محدوده های طبقه بندی را می توان نسبتاً راحت محاسبه کرد و به تجزیه و ررونوی ^۱ داده های آموزشی ارتباط داد. طبق تجزیه و ررونوی هر نقطه x_i یک محدوده دارد که

^۱Voronoi Tessellation



شکل ۲-۲: کلاس طبقه بندی شده با ۱ همسایگی

منطقه را برای آن نزدیکترین نقطه ورود محدود می کند [۱].

می توان به این نتیجه رسید که در داده های آموزشی، بهترین مقدار برای k ، ۱ است. اما با ورود داده جدید دچار مشکل بیش برآزش می شود و همچنین خطا افزایش می یابد. حال اگر این داده های جدید به تعداد فراوانی باشند این مقدار خطا بسیار چشمگیر است.

مسئله بهترین انطباق یا نزدیکترین همسایه برای نمونه های داده ای است که نمونه ها را با ویژگی های ارزشمند واقعی ذخیره می کند. مسئله این است که نمونه هایی را که بیشتر به یک نمونه پرس و جو بر اساس برخی از اندازه گیری ها نزدیک هستند، پیدا می کند.

مثلا ممکن است یک فایل داده حاوی اطلاعات در همه شهرهای با ادارات پست باشد. وابسته به هر شهرستان طول و عرض جغرافیایی آن متفاوت است. اگر نامه ای به یک شهر بدون دفتر پست ارسال شده باشد، نزدیکترین شهر که اداره پست را دارد ممکن است به عنوان مقصد انتخاب شود.

ساده ترین راه حل برای یافتن نزدیکترین همسایگی استفاده از جستجوی سراسری است که تمام رکوردها را مشاهده می کند. اما این روش بسیار زمان بر و پر هزینه می باشد و استفاده از این روش مقرون به صرفه نیست.

فینکل^۱ و بنتلی^۲ [۱] یک ساختار درختی را توصیف می کنند که برای ذخیره سازی ترکیب کلیدها می باشد. این یک تعمیم از درخت دودویی برای ذخیره داده ها در کلید های تک است. بنتلی یک تعمیم متفاوتی از یک ساختار یک بعدی دارد. آن را درخت KD نامگذاری کرده است. بنتلی در مقاله خود، پیشنهاد می کند که درختان KD می توانند به بهترین

^۱Finkel ^۲ Bentley

روش حل برسند.

۲-۲ درخت چندبعدی (KD-Tree)

در علوم کامپیوتر، یک درخت KD (کوتاه شده درخت چند بعدی) یک ساختار داده افزاینده برای فضای برای سازماندهی نقاط در یک فضای چند بعدی است. درخت چند بعدی برای برنامه های مختلف مانند جستجوهای که شامل یک کلید جستجوی چندبعدی هستند، مفید است. (مانند جستجوهای محدوده و جستجوهای نزدیکترین همسایگی). درخت چند بعدی یک مورد خاص از درختان افزاینده فضای دودویی است.

درخت KD یک درخت باینری است که در آن هر گره یک نقطه K بعدی است. هر گره غیر برگ مولد یک ابرصفحه می باشد که فضا را به دو نیمه تقسیم می کند که $half-spaces$ نامیده می شود. نقاطی که در سمت چپ ابرصفحه^۱ قرار دارند، توسط زیر درخت سمت چپ آن گره نشان داده می شوند و نقاط سمت راست ابرصفحه توسط زیر درخت سمت راست نشان داده می شوند. جهت ابرصفحه به صورت زیر تعیین می شود: هر گرهی که با یکی از ابعاد درخت در ارتباط باشد در یک ابرصفحه عمود بر محور قرار دارد. بنابراین؛ برای مثال، اگر در یک تقسیم خاص، محور x انتخاب شود، تمام نقاط با مقدار کوچکتر از x از گره در زیر درخت سمت چپ و تمام نقاط با مقدار بزرگتر از x در زیر درخت سمت راست ظاهر می شوند. در چنین مواردی سطح ابرصفحه توسط مقدار x آن نقطه تعیین می شود و واحد محور x نرمال می شود [۲].

از آنجایی که راه های زیادی برای انتخاب خط تقسیم محور وجود دارد، راه های زیادی هم برای ساخت درخت وجود دارد. یک روش استاندارد ساخت درخت دارای محدودیت های زیر است: برای ساخت یک درخت، یک چرخه از طریق محورها برای انتخاب صفحه تقسیم کننده بکار گرفته می شود (به عنوان مثال ابتدا ریشه یک صفحه همتراز با سطح x دارد، کودکان ریشه دو صفحه همتراز با محور y دارند، نوه های ریشه صفحات همتراز با محور z دارند، فرزند نوه های ریشه محور همتراز با x دارند، نوه های نوه های ریشه محور همتراز با y دارند و به همین صورت). نقاط، با انتخاب میانه نقاط موجود، در زیر درخت قرار داده می شوند. البته با توجه به مختصاتشان در محور مورد استفاده برای ایجاد صفحه تقسیم بندی. این روش منجر به ساخت درخت متوازن می شود که گره ها با ریشه در یک فاصله هستند. البته برای همه برنامه ها درخت متوازن مطلوب نمی باشد.

انتخاب نقطه میانه باعث متوازن شدن درخت می شود. برای جلوگیری از پیچیدگی زمانی می توان از الگوریتم یافتن میانه در [۲] و [۳] یا الگوریتم های مرتب سازی مانند مرتب سازی هر می^۲ یا مرتب سازی ادغامی^۳ یا مرتب سازی های دیگر استفاده کرد. می توان ابتدا تعدادی از نقاط را به صورت تصادفی انتخاب و مرتب کرد و سپس نقطه میانه را یافت تا به عنوان صفحه تقسیم بندی انتخاب شود. این عمل اغلب در درختان متوازن تر نتیجه می دهد.

قابل ذکر است که مقادیر بعد از میانه همگی مقدارشان از میانه بیشتر است. برای نقاطی که روی میانه قرار دارند می توان یک سوپر کلید در نظر گرفت که تمام نقاط را در تمام ابعاد مقایسه می کند. یا اینکه این نقاط مساوی با میانه را در یک طرف

^۱hyperplane

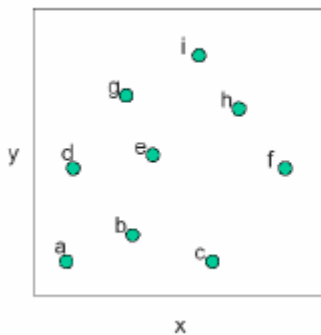
^۲Heap Sort

^۳Merge Sort

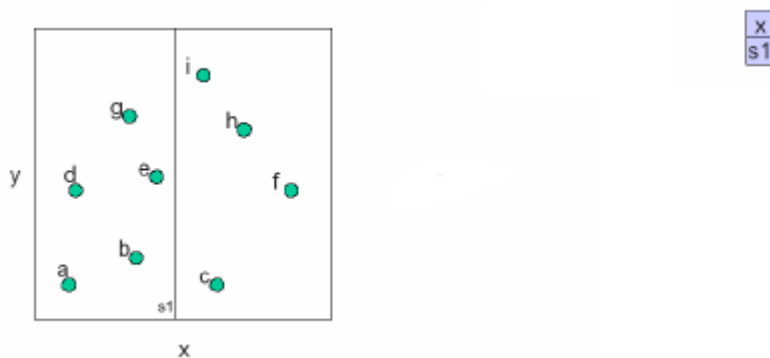
میانۀ قرار داد مثلا نقاط کمتر از میانۀ و نقاط بیشتر مساوی با میانۀ.

الگوریتم دیگر برای ساخت یک درخت چندبعدی متوازن، اطلاعات را قبل از ساخت درخت مرتب می‌کند. سپس این مرتب سازی را در طی ساخت درخت حفظ می‌کند. به این ترتیب گام پرهزینه پیدا کردن میانۀ در هر سطح تقسیم بندی از بین می‌رود. این دو الگوریتم یک درخت متوازن چند بعدی را با مرتب سازی مثلی برای بهبود زمان اجرا ایجاد می‌کند. این الگوریتم های پرسورت قبل از ساخت درخت چندبعدی، درخت را در زمان اجرای $O(n \log n)$ در بهترین حالت می‌سازند. در طول ساخت درخت، نظم و ترتیب این پرسورت ها حفظ می‌شود و از این طریق، در هر سطحی از تقسیم، پیدا کردن میانۀ حذف می‌شود.

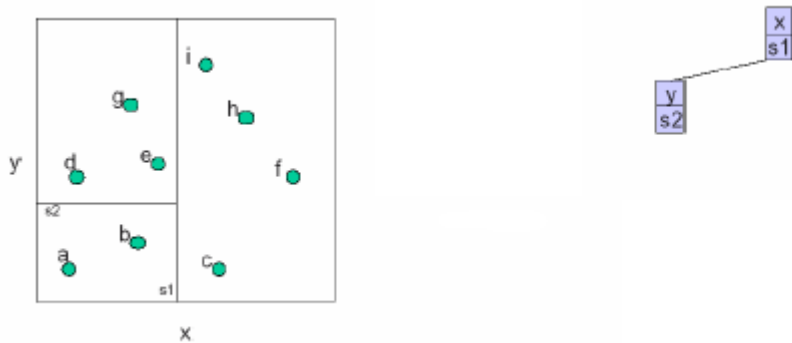
در تصاویر؟؟ نحوه ساخت یک درخت بر روی تعدادی داده نشان داده شده است.



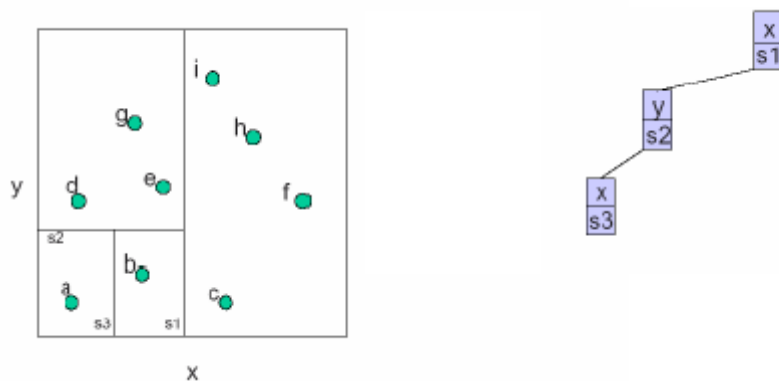
(آ) مرحله اول ساخت درخت: نمایش داده های ورودی جهت ساخت درخت



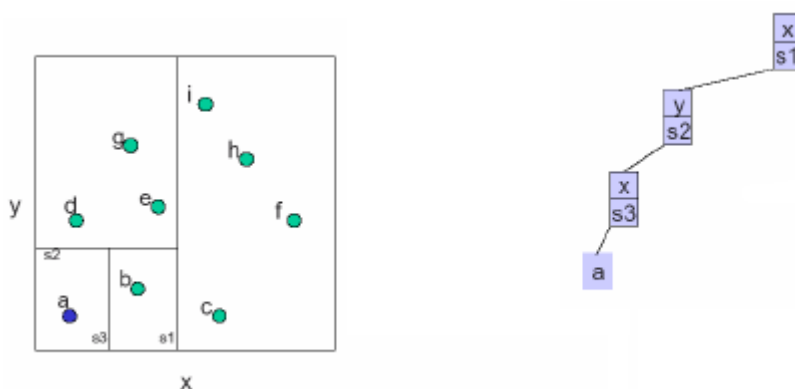
(ب) مرحله دوم ساخت درخت: میزان پراکندگی در هر بعد محاسبه می‌شود و محوری که بیشترین میزان پراکندگی داده ها را دارد انتخاب و میانۀ داده به عنوان خط تقسیم انتخاب می‌شود. در اینجا میزان پراکندگی در جهت محور X بیشتر است. خط تقسیم، s_1 و به عنوان گره ریشه در نظر گرفته می‌شود.



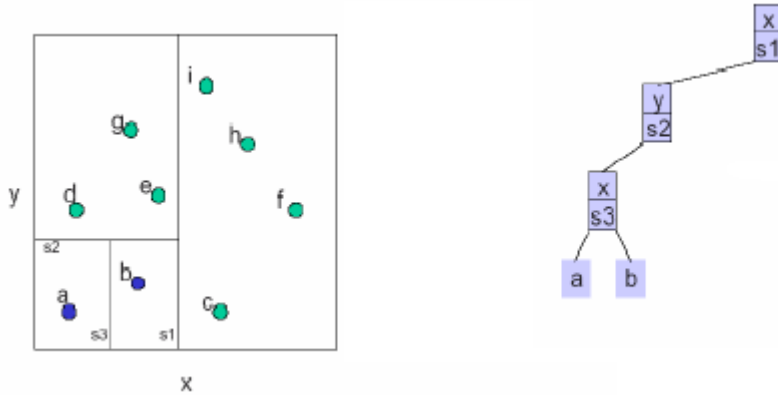
(ج) مرحله سوم ساخت درخت: ابتدا سمت چپ ریشه شروع به ساخت می‌شود. میزان پراکندگی در سمت چپ در جهت محور Y بیشتر است در نتیجه خط تقسیم در جهت محور Y و s_2 در نظر گرفته می‌شود.



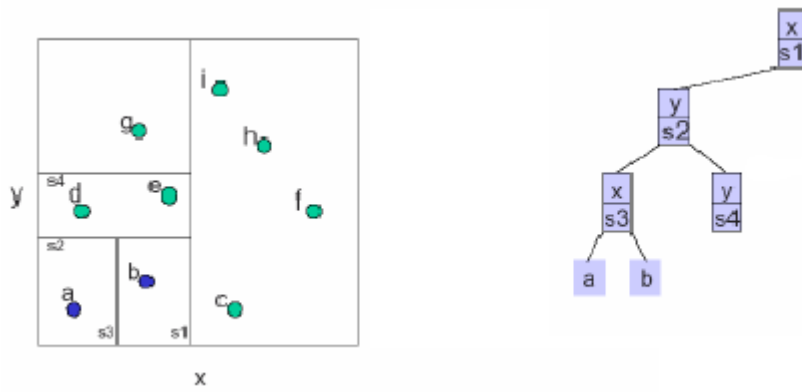
(د) مرحله چهارم ساخت درخت: در سمت چپ s_2 بیشترین میزان پراکندگی در جهت محور X است. پس میانه داده های این محدوده به عنوان خط تقسیم s_3 در نظر گرفته می‌شود.



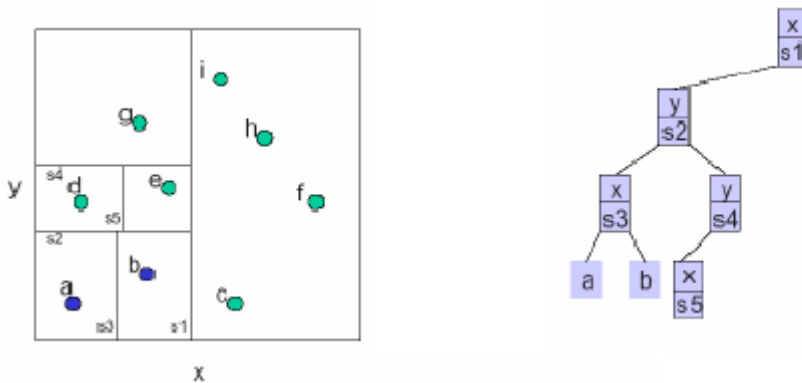
(ه) مرحله پنجم ساخت درخت: سمت چپ خط تقسیم s_3 فقط یک داده a وجود دارد پس به عنوان برگ درخت معرفی می‌شود.



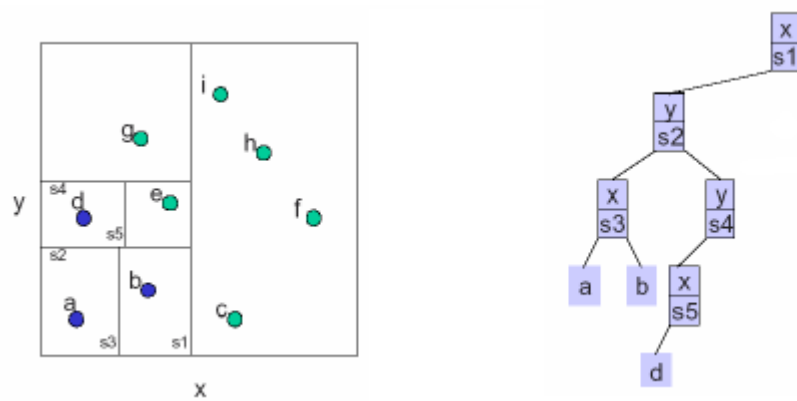
(و) مرحله ششم ساخت درخت: سمت راست خط تقسیم s_3 فقط یک داده b وجود دارد پس به عنوان برگ درخت معرفی می‌شود.



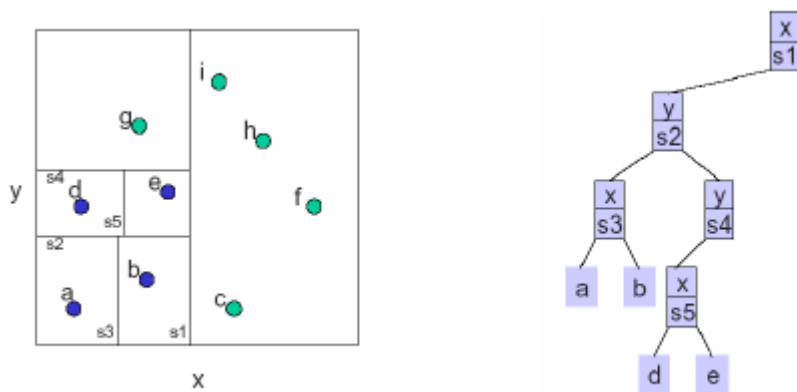
(ز) مرحله هفتم ساخت درخت: سمت راست خط s_2 در نظر گرفته می‌شود. در این ناحیه بیشترین میزان پراکندگی داده‌ها در جهت محور Y است پس s_4 به عنوان خط تقسیم این ناحیه معرفی می‌شود.



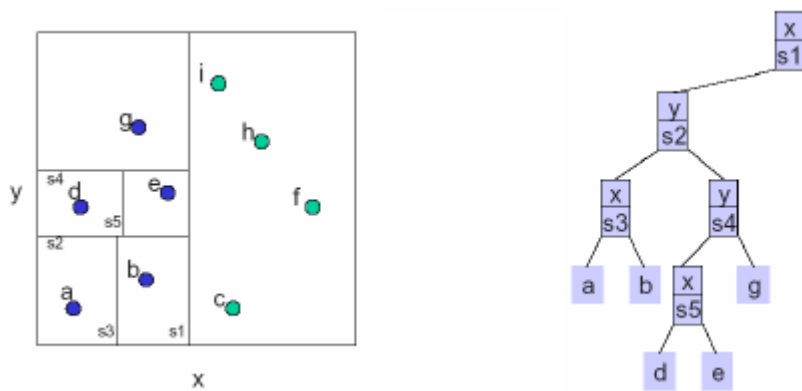
(ح) مرحله هشتم ساخت درخت: در سمت چپ s_4 دو داده وجود دارد که عمل تقسیم بندی در جهت محور X انجام می‌شود و خط تقسیم s_5 معرفی می‌شود.



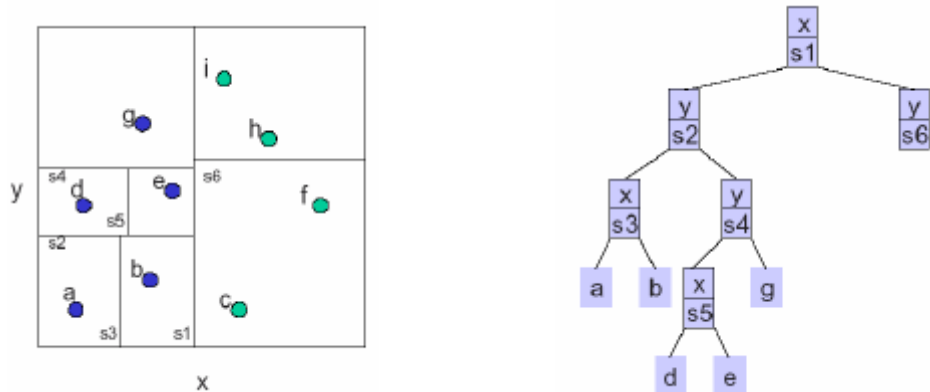
(ط) مرحله نهم ساخت درخت: در سمت چپ خط s_5 داده d وجود دارد که به عنوان برگ معرفی می‌شود.



(ی) مرحله دهم ساخت درخت: سمت راست خط s_5 داده e به تنهایی وجود دارد که به عنوان برگ معرفی می‌شود.



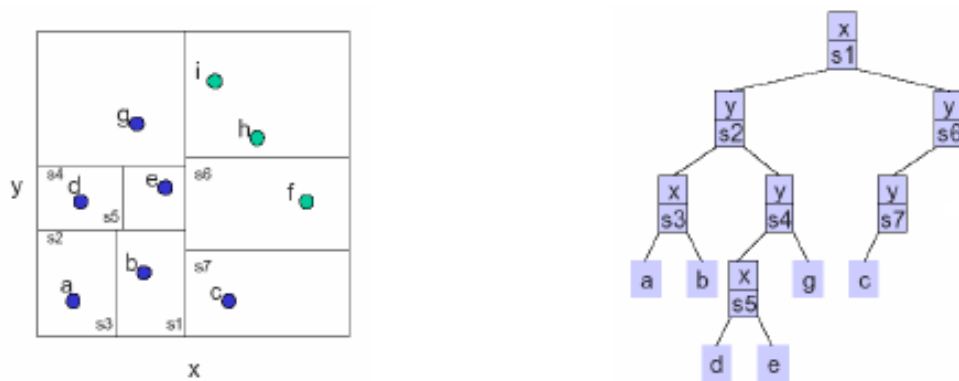
(ک) مرحله یازدهم ساخت درخت: در سمت راست خط s_4 داده g به تنهایی وجود دارد که به عنوان برگ معرفی می‌شود.



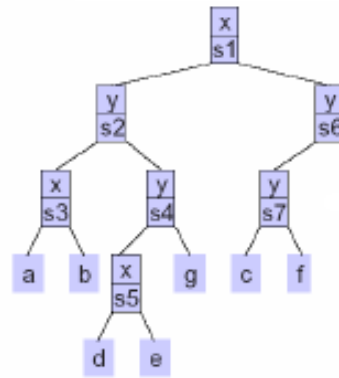
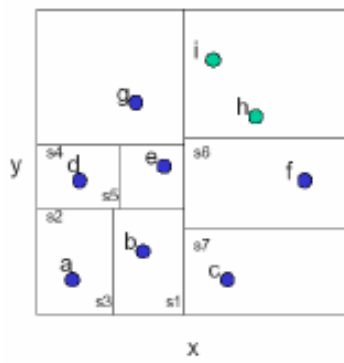
(ل) مرحله دوازدهم ساخت درخت: سمت چپ گره ریشه تکمیل شد. اکنون سمت راست آن باید ساخته شود. در سمت راست خط s_1 میزان پراکندگی داده ها در جهت محور Y بیشتر است پس میانه داده ها در جهت محور Y به عنوان خط تقسیم s_6 معرفی می شود.



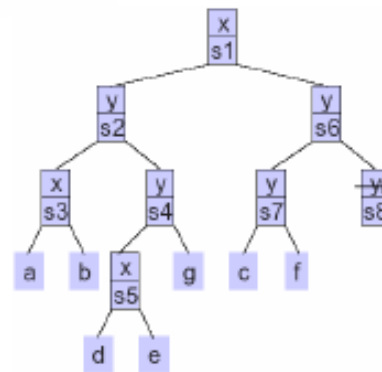
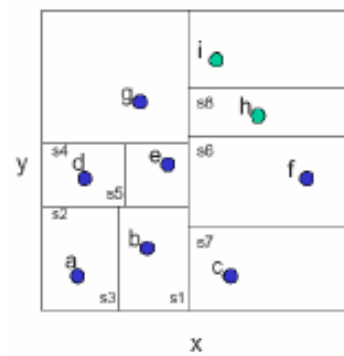
(م) مرحله سیزدهم ساخت درخت: در سمت چپ خط تقسیم s_6 میزان پراکندگی داده ها در جهت محور Y بیشتر است. میانه داده ها در جهت محور Y به عنوان خط تقسیم s_7 معرفی می شود.



(ن) مرحله چهاردهم ساخت درخت: در سمت چپ خط تقسیم s_7 داده c به تنهایی وجود دارد پس به عنوان برگ معرفی می شود.



(س) مرحله پانزدهم ساخت درخت: در سمت راست خط تقسیم s_7 داده f به تنهایی وجود دارد پس به عنوان برگ معرفی می شود.



(ع) مرحله شانزدهم ساخت درخت: در ناحیه سمت راست خط تقسیم s_7 میزان پراکندگی داده ها در جهت محور Y بیشتر است. میانه داده ها به عنوان خط تقسیم s_8 معرفی می شود.



(ف) مرحله آخر ساخت درخت: در سمت چپ خط تقسیم s_8 داده h به تنهایی وجود دارد پس به عنوان برگ معرفی می‌شود. همچنین در سمت راست خط تقسیم s_8 داده i به تنهایی وجود دارد پس به عنوان برگ معرفی می‌شود. در این مرحله درخت تکمیل شد.

شکل ۲-۳: نحوه ساخت درخت KD Tree

۱-۲-۲ اضافه کردن داده به درخت

همانطور که به درخت جستجوی یک عنصر اضافه می‌شود، به درخت چندبعدی هم داده اضافه می‌شود. برای اضافه کردن نقطه، از ریشه شروع کنید و درخت را ببینید. بسته به اینکه نقطه ای که قرار است روی سمت چپ یا راست قرار بگیرد به طرف سمت چپ یا راست حرکت کنید. با توجه به محور تقسیم بندی نقطه را با گره مقایسه کنید. با فرض مثال محور تقسیم در ریشه x است پس مولفه x نقطه جدید را با مولفه x ریشه مقایسه کنید. اگر بیشتر بود به سمت راست و اگر کمتر بود به سمت چپ حرکت کنید. اضافه کردن نقطه باعث نامتوازن شدن درخت و کاهش میزان عملکرد درخت می‌شود. میزان تضعیف عملکرد درخت بستگی به توزیع فضایی نقاط اضافه شده و تعداد نقاط اضافه شده دارد. اگر یک درخت بیش از حد نامتوازن باشد باید دوباره متوازن شود تا عملکرد نمایش داده‌ها که به توازن درخت تکیه دارند مانند جستجوی نزدیک‌ترین همسایه بالا رود [۴].

۲-۲-۲ حذف یک گره

برای حذف یک گره از یک درخت چند بعدی، ساده‌ترین روش این است که آن گره را حذف و مجموعه برگ‌ها و گره‌های فرزندان آن گره را دوباره بازسازی کنید.

یک رویکرد دیگر این است که جایگزینی برای نقطه پیدا کنید ([۴]). ابتدا گره R را که شامل نقطه ای است که باید حذف شود پیدا کنید. در مواردی که گره R یک برگ است جایگزین لازم نیست. ولی در مواردی غیر از این، از زیر درخت یک نقطه را انتخاب و به نام P جایگزین کنید و سپس P را حذف کنید. برای پیدا کردن نقطه جایگزین، اگر R در x باشد

و R دارای فرزند راست باشد، نقطه را با مقدار حداقل X از زیر درخت ریشه در فرزند راست پیدا کنید. در غیر این صورت، نقطه را با حداکثر مقدار X از زیر درخت ریشه در فرزند سمت چپ پیدا کنید. تعادل یک درخت چند بعدی نیازمند احتیاط است زیرا درخت چند بعدی در ابعاد مختلف طبقه بندی می شود. بنابراین، روش چرخش درخت، نمی تواند برای تعادل آن استفاده شود، زیرا ممکن است ثبات را از بین ببرد.

۳-۲-۲ جستجوی نزدیکترین همسایه

هدف الگوریتم جستجوی نزدیکترین همسایه، پیدا کردن نزدیکترین نقطه به نقطه ورودی داده شده است. این جستجو می تواند با استفاده از خواص درخت به طور موثر و با سرعت، بخش های زیادی از فضای جستجو را حذف کند.

جستجوی نزدیک ترین همسایه در یک درخت KD به شرح زیر است:

با شروع از گره ریشه، الگوریتم به صورت بازگشتی به سمت درخت حرکت می کند، به همان شیوه که اگر نقطه جستجو وارد شده باشد (به عنوان مثال آن را به سمت چپ یا راست بسته به اینکه آیا نقطه کمتر یا بیشتر از گره فعلی در بعد تقسیم است یا خیر) می رود. هنگامی که الگوریتم به گره برگ برسد، آن نقطه گره را به عنوان "بهترین جریان" ذخیره می کند. الگوریتم فراخوانی، درخت را فرا می خواند و مراحل زیر را در هر گره انجام می دهد [۴]:

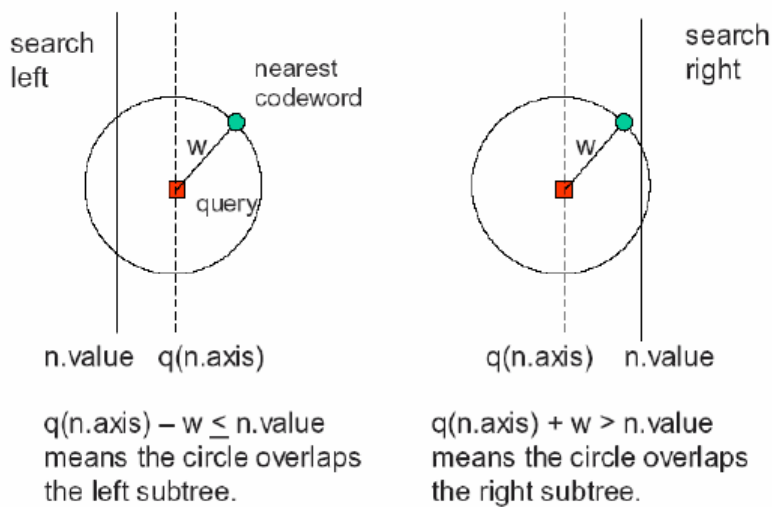
الگوریتم ۱-۲ الگوریتم جستجوی نزدیکترین همسایه با KD Tree.

ورودی: نقطه پرس و جو q

خروجی: نزدیکترین همسایگی

- ۱: موقعیت مناسب q در درخت یافت می شود و داده ای که در آن محدود قرار دارد، گره n نامیده می شود.
- ۲: داده ای که در محدوده q قرار دارد اگر سمت چپ و راست آن خالی باشد یعنی یک برگ است آنگاه میزان $w = ||q - n.point||$ که فاصله q با گره n است، محاسبه می شود. در ابتدا این فاصله به عنوان بهترین فاصله حفظ می شود.
- ۳: اگر گره خالی نبود آن گاه مقدار گره در محور مربوطه اش و مقدار q در جهت محور گره باهم مقایسه می شود. اگر مقدار گره از مقدار q بیشتر باشد به مرحله ۴ در غیر این صورت به مرحله ۶ بروید
- ۴: در صورتی که جستجو از سمت چپ گره شروع شود، اگر مقدار q در جهت محور گره منهای فاصله w کمتر از مقدار گره بود آن گاه به مرحله ۲ بروید و در محدوده سمت چپ گره جستجو را ادامه دهید بعد به مرحله ۶ بروید
- ۵: اگر مقدار q در جهت محور پره به علاوه مقدار w بیشتر از مقدار گره بود به مرحله ۲ بروید و جستجو را در سمت راست گره ادامه دهید بعد به مرحله ۷ بروید
- ۶: اگر مقدار q در جهت محور گره منهای w کمتر از مقدار گره بود به مرحله ۲ بروید و جستجو را در چپ گره ادامه دهید.
- ۷: این مراحل را تا رسیدن به ریشه ادامه دهید.

زمانیکه الگوریتم فرآیند را برای گره ریشه انجام دهد، جستجو کامل می شود. به طور کلی، الگوریتم از فاصله مربعات



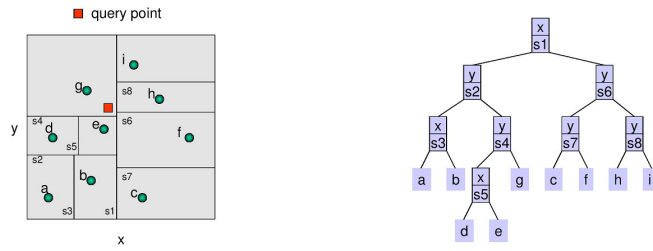
شکل ۲-۴: الگوریتم مصور جستجو در درخت KD Tree

برای مقایسه استفاده می‌کند تا از محاسبه ریشه های مربع جلوگیری شود. علاوه براین، می‌تواند محاسبات را با نگه داشتن بهترین فاصله مربع در یک متغیر، برای مقایسه حفظ کند. در شکل ؟؟ الگوریتم جستجو به صورت مصور نشان داده شده است.

پیچیدگی زمانی این عملیات در مورد نقاط به طور تصادفی توزیع شده برابر با $O(\log n)$ است. این الگوریتم می‌تواند بجای یک نقطه، k نزدیکترین همسایه را بوسیله k بهترین نقاط جاری، حفظ کند. یک شاخه زمانی حذف می‌شود که k نقطه پیدا شود و شاخه نتواند یک نقطه نزدیکتر از k بهترین نقاط جاری داشته باشد. همچنین می‌تواند به یک الگوریتم تقریبی تبدیل شود تا سریعتر اجرا شود. به عنوان مثال، تقریباً نزدیکترین همسایه جستجو را می‌توان به سادگی با تنظیم ماکزیمم محدودیت از تعداد نقاط برای بررسی در درخت به دست می‌آورد.

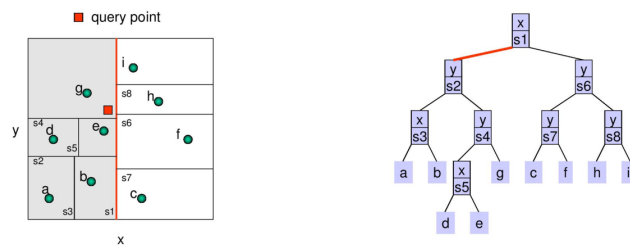
نزدیکترین همسایگی برای نقاطی که فاصله شان از نتیجه صفر است می‌تواند بروزرسانی نشود. این سیر نزولی دور انداختن نقاط منحصر بفرد نیست اما در همسایگی با نقطه جستجوی اصلی است.

در تصاویر ؟؟ پیاده سازی نحوه جستجوی روش درخت چند بعدی بر روی یک مثال نشان داده شده است.



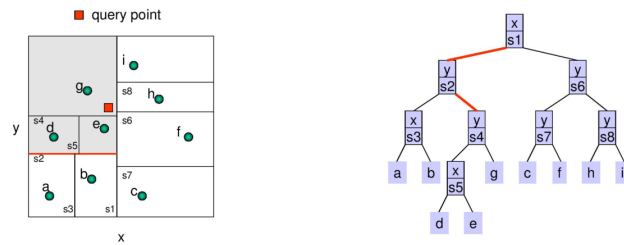
(آ) مرحله اول جستجو در درخت: جستجوی بهترین موقعیت برای query point و

شروع از ریشه



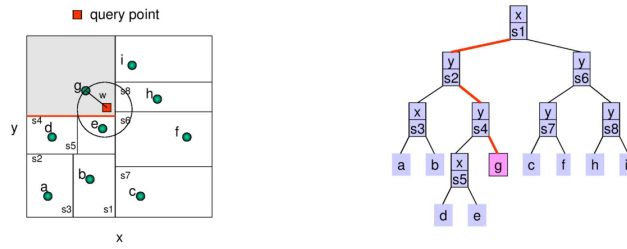
(ب) مرحله دوم جستجو در درخت: مولفه x نقطه query point از خط تقسیم

$s-1$ کمتر است پس محدوده سمت چپ $s-1$ در نظر گرفته می شود.

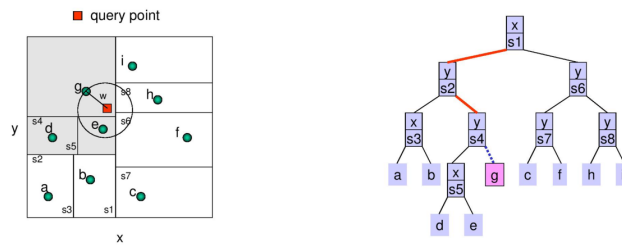


(ج) مرحله سوم جستجو در درخت: مولفه دوم نقطه پرس و جواز خط تقسیم $s-2$

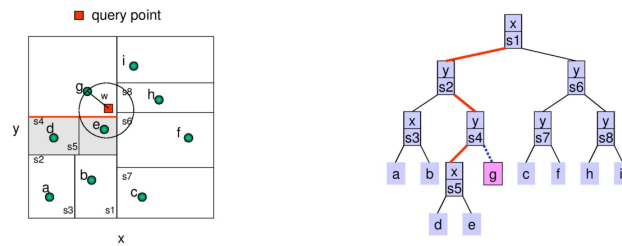
بیشتر است پس محدوده سمت راست $s-2$ در نظر گرفته می شود.



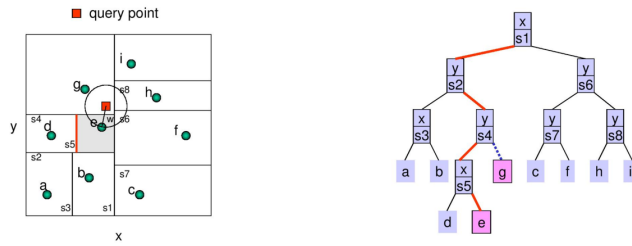
(د) مرحله چهارم جستجو در درخت: مولفه دوم نقطه پرس و جو از خط تقسیم $s - 4$ بیشتر است پس محدوده سمت راست $s - 4$ در نظر گرفته می شود. محدوده نقطه پرس و جو یافت می شود و فاصله query point با g به عنوان w اندازه گیری می شود.



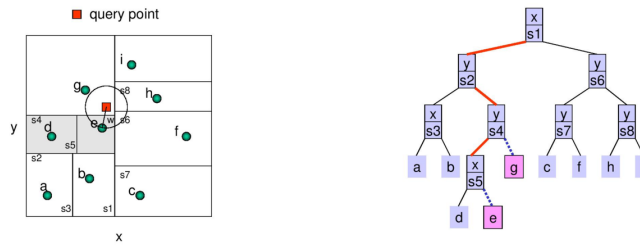
(ه) مرحله پنجم جستجو در درخت: مقدار گره $s - 4$ در جهت محور Y کمتر از مولفه دوم query point است پس ابتدا جستجو باید در سمت راست گره $s - 4$ بررسی شود که این بررسی قبلا انجام شده است.



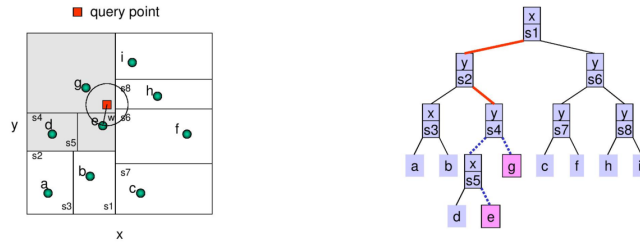
(و) مرحله ششم جستجو در درخت: فاصله گره $s - 4$ از مولفه دوم query point کمتر از w پس جستجو در سمت چپ گره ادامه می یابد.



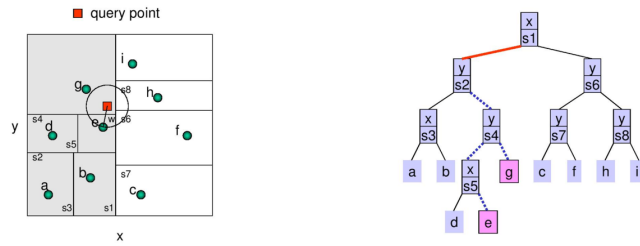
(ز) مرحله هفتم جستجو در درخت: خط تقسیم $s-5$ به عنوان گره معرفی می شود. مولفه اول query point از مقدار $s-5$ بیشتر است پس ابتدا سمت راست بررسی می شود. فاصله query point با e کمتر از w است این فاصله در w قرار می گیرد.



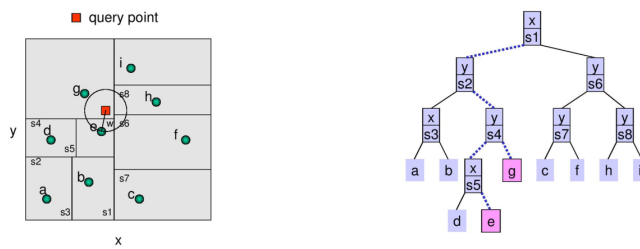
(ح) مرحله هشتم جستجو در درخت: مولفه اول query point منهای w از $s-5$ بیشتر است پس سمت چپ بررسی نمی شود و حذف می شود.



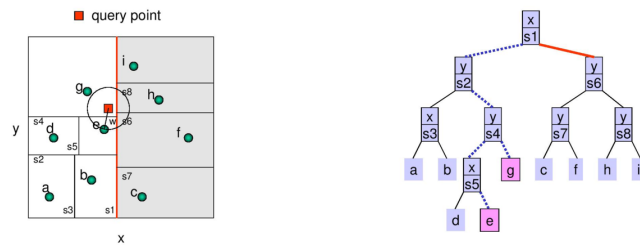
(ط) مرحله نهم جستجو در درخت: به گره $s-2$ برگردید. سمت راست $s-2$ بررسی شده است.



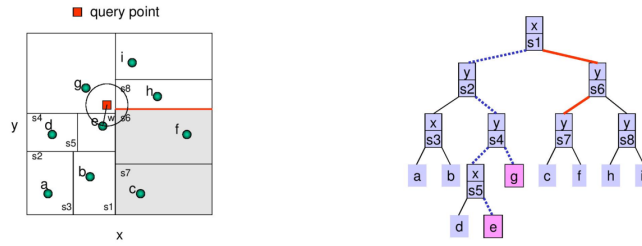
(ی) مرحله دهم جستجو در درخت: مولفه دوم query point منهای w از $s - 2$ بیشتر است پس سمت چپ $s - 2$ بررسی نمی شود و حذف می شود.



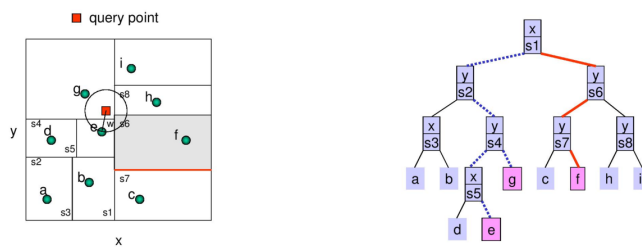
(ک) مرحله یازدهم جستجو در درخت: به گره $s - 1$ بروید. سمت چپ بررسی شده است



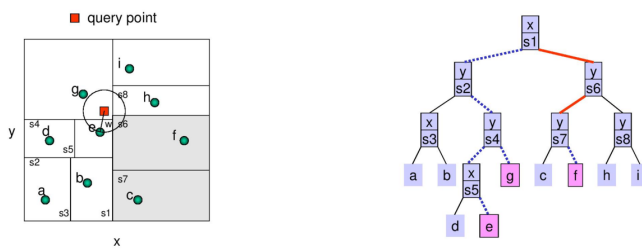
(ل) مرحله دوازدهم جستجو در درخت: مولفه اول query point منهای w کمتر از $s - 1$ است پس باید سمت راست را نیز بررسی کرد.



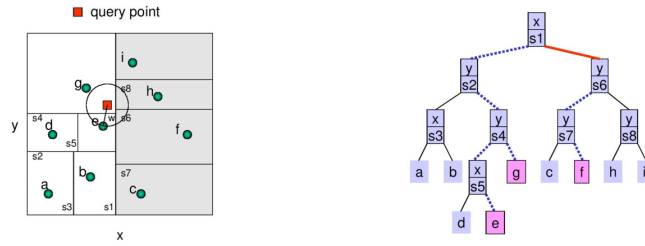
(م) مرحله سیزدهم جستجو در درخت: گره $s - 6$ معرفی می‌شود. جهت $s - 6$ محور Y است. query point بیشتر از $s - 6$ است و query point منهای w کمتر از $s - 6$ است پس باید ابتدا سمت چپ $s - 6$ بررسی شود.



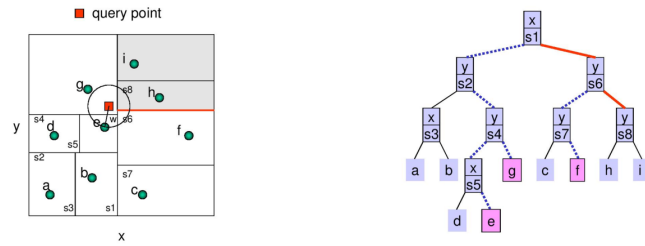
(ن) مرحله چهاردهم جستجو در درخت: سمت چپ $s - 6$ گره $s - 7$ وجود دارد که در جهت محور Y است. query point از $s - 7$ بیشتر و query point منهای w از $s - 6$ بیشتر پس فقط سمت راست $s - 7$ بررسی می‌شود. در این محدوده تنها داده f وجود دارد که فاصله آن با query point بیشتر از w است پس، از آن صرف نظر می‌شود.



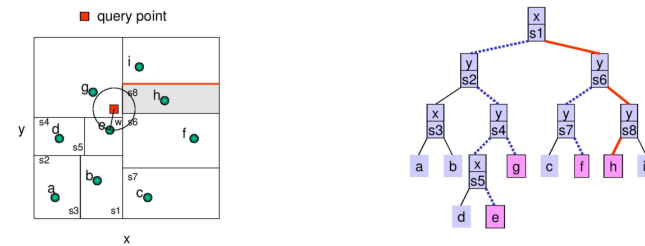
(س) مرحله پانزدهم جستجو در درخت: برگشت به گره $s - 7$.



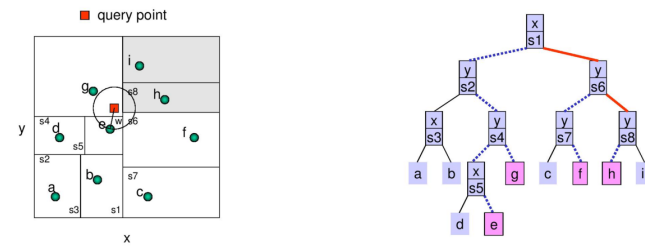
(ع) مرحله شانزدهم جستجو در درخت: برگشت به گره $s - 6$



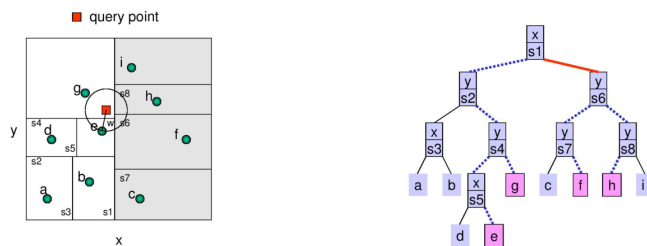
(ف) مرحله هفدهم جستجو در درخت: سمت راست $s - 6$ بررسی می شود.



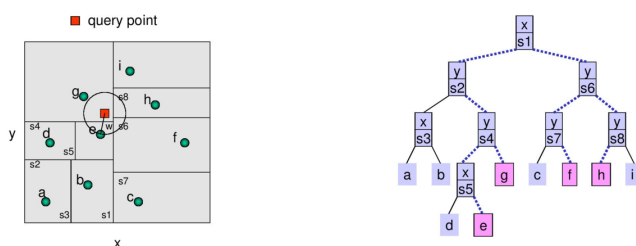
(ص) مرحله هجدهم جستجو در درخت: گره $s - 8$ در جهت محور Y معرفی می شود. query point کمتر از $s - 8$ است و query point به علاوه w کمتر از $s - 8$ است. پس فقط سمت چپ $s - 8$ بررسی می شود که داده h به تنهایی در این محدوده وجود دارد و فاصله h از query point بیشتر از w است پس از این داده صرف نظر می شود.



(ق) مرحله نوزدهم جستجو در درخت: بازگشت به گره $s - 8$



(ر) مرحله بیستم جستجو در درخت: بازگشت به گره ۶ - s



(ش) نتیجه نهایی جستجو در درخت: بازگشت به ریشه. بررسی درخت تمام شده است و در نتیجه نزدیکترین داده به query point، e است. شاخه هایی از درخت که مورد بررسی قرار گرفته اند با خط چین بررسی شده اند. مابقی شاخه ها حذف می شوند.

شکل ۲-۵: نحوه جستجو در درخت KD Tree

فصل ۳

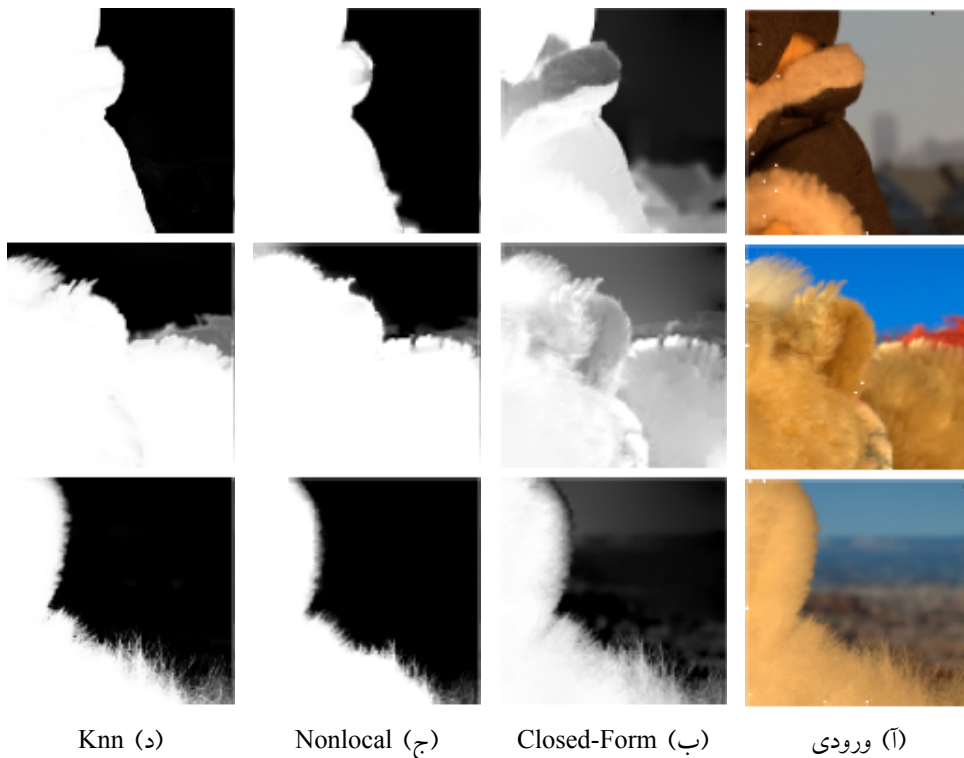
درهم تنیدگی با جستجوی نزدیکترین همسایگی

۱-۳ درهم تنیدگی با جستجوی نزدیکترین همسایگی

در فصل اول با چند روش درهم تنیدگی آشنا شدید. در این فصل یک روش درهم تنیدگی که عملکرد بهتری نسبت به روش های ذکر شده دارد، بیان می شود. نتیجه مقایسه این درهم تنیدگی با روش های دیگر در فصل ۴ بیان شده است. روش های ذکر شده در فصل اول به صورت محلی انجام می شدند. یعنی در هرکدام درهم تنیدگی در پنجره های کوچکی (۳×۳ و ۵×۵) انجام می شد. اما ایراد این روش این است که فرض کنیم تصاویر دارای نویز باشند و یا یک فیلتر بر روی آن ها اعمال شده باشد، آنگاه همان نویز یا فیلتر می تواند رنگ پیکسل های ناشناخته را به خود معطوف کند و به اشتباه درهم تنیدگی را انجام دهد. از این رو روش های درهم تنیدگی غیر محلی پیشنهاد داده شدند. یکی از این روش ها درهم تنیدگی، مبتنی بر نزدیکترین همسایگی^۲ است که به صورت غیر محلی یا سراسری این کار را انجام می دهد. مزیت دیگر این روش این است که تنها با چند کلیک و با استفاده از نشانه گذاری های کاربر می تواند این مسئله را حل کند اما روش های دیگر به دلیل استفاده از فرض محلی بودن نمی توانند به طور قابل اطمینان با چنین نشانه گذاری عمل درهم تنیدگی را انجام دهند. زیرا فقط می توانند رنگ های موجود در پنجره شان را تشخیص دهند و در فواصل بیشتر کارایی مطلوبی ندارند. این مسئله در تصویر؟؟؟ به صورت مصور بیان شده است. در تصویر؟؟؟ با استفاده از کلیک روی تصاویر دور روش فرم بسته و KNN مقایسه شده اند که ظاهراً روش KNN نتیجه بهتری را تولید کرده است. در تصاویر بالا مرزها واضح تر و تمیزتر، در تصویر وسط جزئیات بیشتری برای پرزها و و رنگ قرمز به خوبی جدا شده است و در تصویر پایین خزها از پیش زمینه جدا شده اند.

در مسائل قبلی از یک نقشه سه گانه استفاده می کردیم. که این نقشه به این صورت بود که دارای سه محدوده پیش زمینه، پس زمینه و ناحیه ناشناخته بود. اما در اینجا نیازی به چنین نقشه ای نیست و تنها با چند کلیک می توان عمل جداسازی پیش زمینه از پس زمینه را انجام داد. این در صورتی است که تکنیک های پیشین در نمونه برداری های پراکنده شکست می خورند

^۲KNN Matting



شکل ۱-۳: مقایسه روش های درهم تنیدگی فرم بسته، غیر محلی و نزدیکترین همسایگی

(در فصل یک در روش بیز این مسئله توضیح داده شده است) ولی در این روش اینگونه نیست و ورودی های کاربر می توانند با فاصله های زیادی انتخاب شوند.

روش های پیشین تنها در یک یا چند مدل رنگی خاص (RGB, HSV, ...) کار می کنند، مانند روش فرم بسته که در مدل رنگی RGB تعریف شده است. روش درهم تنیدگی نزدیکترین همسایگی با هر مدل رنگی به خوبی کنار می آید و در واقع از هیچ مدل رنگی خاصی پیروی نمی کند.

مزیت اصلی این روش این است که برای استخراج چند لایه از تصویر به کار می رود. یعنی اینکه می توان چند ناحیه را به طور همزمان به صورت جداگانه شناسایی کرد. در فصل اول در؟؟ گفتیم که:

$$I = \alpha F + (1 - \alpha)B$$

در واقع اگر پیش زمینه و پس زمینه، هر کدام یک لایه در نظر گرفته شود، ترکیب؟؟ را می توان به صورت زیر نوشت:

$$I = \sum_{i=1}^n \alpha_i F_i \quad (1-3)$$

که

$$\sum_{i=1}^n \alpha_i = 1 \quad (2-3)$$

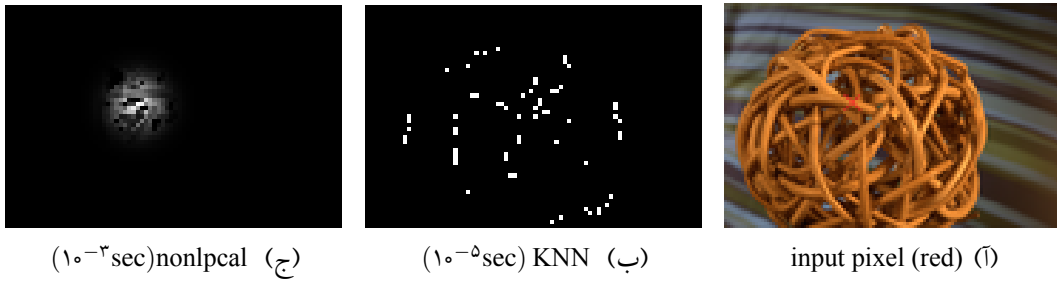
و F_i مربوط به پیش زمینه i است. در واقع ترکیب؟؟ بجای اینکه به صورت یک پیش زمینه و یک پس زمینه تعریف شود، به صورت چند پیش زمینه تعریف می شود.

۳-۱-۱ محاسبه A با نزدیکترین همسایگی

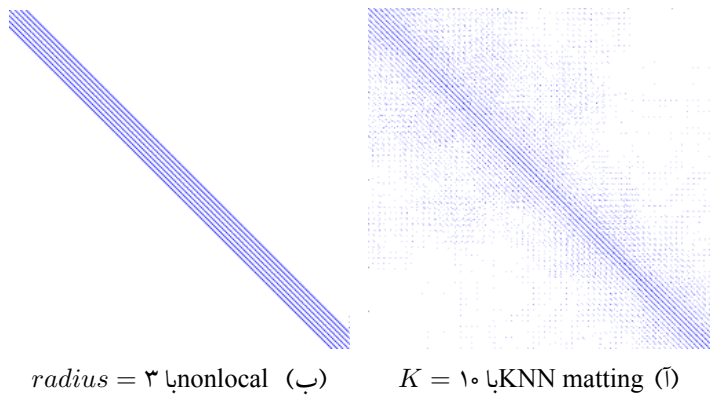
در فصل اول در بخش مدل رنگ سراسری در مورد درهم تنیدگی غیر محلی صحبت شد. طبق معادله؟؟ مربوط به درهم تنیدگی غیر محلی A یک تابع هسته است که مجموع وزن داری از پیکسل های اطراف یک پیکسل مشخص را معرفی می کند. محاسبه A در درهم تنیدگی نزدیکترین همسایگی شامل جمع آوری همسایگان غیر محلی از پیکسل I است البته قبل از اینکه بردارهای ویژگی X با استفاده از $K(i, j)$ تخمین زده شوند. در اینجا به جای استفاده از یک تابع کرنل بزرگ، با توجه به یک پیکسل i با استفاده از محاسبه K نزدیکترین همسایگی در فضای ویژگی، اصل غیر محلی اجرا می شود. مشکل اصلی درهم تنیدگی غیر محلی بار محاسباتی تابع کرنل بود. اما در تکنیک درهم تنیدگی نزدیکترین همسایگی استفاده از تابع هسته به علت جستجو با روش knn مطرح نیست و در نتیجه زمان اجرا برای محاسبه یک ردیف از A برابر است با $O(kq)$ که $O(q)$ برابر با زمان هر پرس و جو در FLANN است.

تعداد همسایگی (K) معمولاً بین ۳ تا ۱۵ است که ۳ برای تصاویر اشیاء و ۱۵ برای تصاویر طبیعت است. دلیل این مقادیر می تواند این باشد که تصاویر طبیعت دارای بافت پیچیده تری هستند و برای یافتن جواب نیاز به همسایگان بیشتری است.

در شکل؟؟ همسایگی غیر محلی محاسبه شده با درهم تنیدگی نزدیکترین همسایگی و درهم تنیدگی غیر محلی مقایسه شده اند و اثربخشی جستجوی KNN در فضای ویژگی در اجرای اصل غیر محلی نشان داده شده است. برای این کار از یک تابع هسته با شعاع ۹ استفاده شده است. در درهم تنیدگی نزدیکترین همسایگی بجای استفاده از این تابع هسته از $K = ۸۱$ استفاده شده است. مشاهده می شود که زمان اجرا درهم تنیدگی نزدیکترین همسایگی از درهم تنیدگی غیر محلی کمتر است. تصویر؟؟ محاسبات معمولی A را در درهم تنیدگی نزدیکترین همسایگی به تصویر در می آورد. طبق این تصویر روش درهم تنیدگی نزدیکترین همسایگی همسایگان بیشتری را پیدا کرده است و عمل غیر محلی را نسبت به درهم تنیدگی غیر محلی بهتر انجام داده است.



شکل ۳-۲: مقایسه KNN و nonlocal



شکل ۳-۳: محاسبات A در KNN matting

۲-۱-۳ بردار ویژگی x با مختصات مکانی

درهم تنیدگی نزدیکترین همسایگی می تواند بجز فضای رنگی RGB به راحتی برای رسیدگی به فضاهای رنگی دیگری (مانند: HSV) گسترش داده شود. برای درهم تنیدگی طبیعی، بردار ویژگی $X(\cdot)$ در یک پیکسل معین i که شامل مختصات مکانی برای تقویت همبستگی مکانی است، در فضای رنگی HSV به صورت زیر می تواند تعریف شود:

$$X(i) = (\cos(h), \sin(h), s, v, x, y)_i \quad (۳-۳)$$

که h, s, v مختصات HSV و (x, y) مختصات مکانی پیکسل i هستند. بردار ویژگی $X(\cdot)$ در فضای رنگی RGB به صورت زیر است:

$$X(i) = (r, g, b, x, y)_i \quad (۴-۳)$$

که r, g, b سطوح رنگ یا مختصات RGB و (x, y) مختصات مکانی پیکسل i هستند.

۳-۱-۳ محاسبه تابع هسته برای تقسیم بندی نرم

در ابتدا تعریفی از تقسیم بندی نرم و تقسیم بندی سخت را ذکر می کنیم. اگر عمل تقسیم بندی به طور مطلق باشد یعنی به طور مطلق تصمیم گرفته شود که منطقه ناشناخته پیش زمینه یا پس زمینه است، به آن تقسیم بندی سخت^۱ و اگر عمل تقسیم بندی با احتمال باشد یعنی اینکه منطقه ناشناخته با یک احتمالی به پیش زمینه یا پس زمینه تعلق داشته، تقسیم بندی نرم^۲ نامیده می شود.

تابع هسته $k(x)$ که برابر است با $1 - x$

$$k(X_i, X_j) = 1 - \frac{\|X(i) - X(j)\|}{C} \quad (۵-۳)$$

که $\|X(i) - X(j)\|$ فاصله دو پیکسل i و j است، را تحلیل می کنیم. در اینجا C کمینه کران بالا از $\|X(i) - X(j)\|$ برای ساختن $k(X_i, X_j) \in [0, 1]$ است. KNN کوچکترین مقدار $\|X(i) - X(j)\|$ یعنی نزدیکترین فاصله بین دو پیکسل i و j را بر می گرداند. یک انتخاب معمولی از هسته ها در یادگیری ماشین $\exp(-x)$ است که در درهم تنیدگی غیر محلی استفاده شده است. همانطور که در شکل ۴؟ نشان داده شده است معادله $\exp(-x)$ تمایل به ایجاد یک تقسیم بندی سخت دارد، زیرا مقدار تابع هسته با توجه به تفاوت رنگ، به طور نمایی کاهش می یابد. بر عکس، اصطلاح $1 - x$ بدون مختصات مکانی می تواند تقسیم بندی نرم را به پس زمینه واقعی نزدیکتر کند. علاوه بر این، با استفاده از $1 - x$ با مختصات مکانی، می توان یک درهم تنیدگی هموارتر بین پیکسل های همسایه ایجاد کرد.

در درهم تنیدگی غیر محلی که در فصل ۱ شرح داده شده است، نویسندگان اشاره کرده اند که خوشه بندی لاپلاسی در موجب خطا در اطراف لبه می شود، در حالیکه دلیل عمده ممکن است استفاده از شرایط تعریف تابع هسته باشد. فرض کنید $F = (1, 0, 0)$ و $B = (0, 0, 0)$ و بردار ویژگی یک پیکسل $E = (0.3, 0, 0)$ است. با استفاده از درهم تنیدگی غیر محلی و بدون در نظر گرفتن شرایط مکانی، میزان تابع هسته، برای مشخص نمودن اینکه پیکسل E پیش زمینه است یا پس زمینه، به صورت زیر محاسبه می شود.

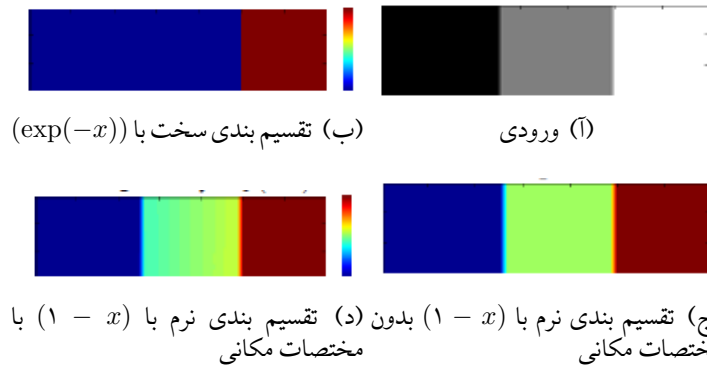
$$k(F, E) = e^{(-\|F-E\|^2/h^2)} = \exp(-0.7^2/0.01) = e^{(-۴۹)}$$

$$\text{and } k(B, E) = \exp(-0.3^2/0.01) = e^{(-۹)}$$

که $k(F, E) \ll k(B, E)$ بسیار کوچکتر از B است، و بنابراین تقسیم بندی سخت حاصل می شود. از نظر عددی، این نیز باعث بی ثباتی در محاسبه خوشه بندی لاپلاسی می شود، زیرا مقدار بسیاری از عبارات بسیار ناچیز و کوچک است. در نتیجه در درهم تنیدگی مبتنی بر نزدیکترین همسایگی بجای اینکه از همان اول از تابع کرنل استفاده کنیم و در همان تابع بردار ویژگی X را محاسبه کنیم، پس از اینکه بردار ویژگی X محاسبه شد از تابع کرنل جهت ادامه فرآیند استفاده می

^۱hard segmentation

^۲soft segmentation



شکل ۳-۴: مقایسه تقسیم بندی نرم و تقسیم بندی سخت

کنیم. اینگونه سرعت کار افزایش می یابد و بار محاسباتی جهت بدست آوردن بردار ویژگی X کاهش می یابد.

۴-۱-۳ پیاده سازی سریع برای روش فرم بسته

از آنجاییکه لاپلاسیان $L_c = (D - A)^T(D - A)$ برای گراف لاپلاسیان مساعد است، ولی لاپلاس $L = D - A$ تنک است و به سرعت اجرا می شود (حدود ۱۰۰ بار سریعتر از L_c). بدون از دست دادن عمومیت، L در این بخش استفاده می شود. برای محاسبه L روش فرم بسته در پیش گرفته می شود. هنگامی که ورودی های کاربر در قالب نقشه سه گانه هستند، طبق رابطه؟؟، می توان نشان داد که راه حل فرم بسته برای استخراج بیشتر از دو لایه به صورت زیر است.

$$(L + \lambda D) \sum_i^n \alpha_i = \lambda m \quad (۶-۳)$$

همانطور که در فصل ۱ ذکر شده است، D یک ماتریس قطری $diag(m)$ و m یک بردار دودویی از همه شاخص های پیکسل های مشخص شده است، و λ یک ثابت کنترل اعتماد کاربر روی نشانه ها است. تابع بهینه $g(x)$ برای فرم بسته که در فصل ۱ در رابطه؟؟ برای α ذکر شده است، برای X در رابطه؟؟ آمده است.

$$g(x) = x^T Lx + \lambda \sum_{i \in m-v} x_i^2 + \lambda \sum_{i \in v} (x_i - 1)^2 \quad (۷-۳)$$

که v یک بردار دودویی از شاخص های پیکسل مربوطه از نشانه گذاری های کاربر برای یک لایه معین است. در نهایت:

$$\begin{aligned}
 g(x) &= x^T Lx + \lambda \sum_{i \in m-v} x_i^2 + \lambda \sum_{i \in v} x_i^2 - 2x \times (1) + (1)^2 \\
 &= x^T Lx + \lambda \sum_{i \in m-v} x_i^2 + \lambda \sum_{i \in v} x_i^2 - 2\lambda v^T x + \lambda|v| \\
 &= x^T Lx + \lambda \sum_{i \in m} x_i^2 - 2\lambda v^T x + \lambda|V| \\
 &= \frac{1}{2} x^T 2(L + \lambda D)x - 2\lambda v^T x + \lambda|V| \\
 &= \frac{1}{2} x^T Hx - c^T x + \lambda|V|
 \end{aligned}$$

که $\lambda|v|$ یک ثابت است. توجه داشته باشید که

$$H = 2(L + \lambda D) \quad (8-3)$$

یک ماتریس نیمه معین مثبت است زیرا L نیمه معین مثبت است و D یک ماتریس قطری تولید شده توسط بردار دودویی m است. مشتق $g(x)$ نسبت به x را برابر صفر قرار می دهیم.

$$\frac{\partial g}{\partial x} = Hx - c = 0 \quad (9-3)$$

که در نهایت:

$$H^{-1}c = (L + \lambda D)^{-1}c \quad (10-3)$$

جهت محاسبه H ، که یک ماتریس تنک و بزرگ و متقارن و نیمه معین مثبت است، می توان از اجرای PCG در [؟] که در واقع یک الگو برای حل سیستم های خطی ارائه می دهد و حدود ۵ ثانیه سریعتر از روش مرسوم گرادیان مزدوج است، به منظور حل تصاویر طبیعی موجود در سایت Alpha Matting استفاده کرد. در نهایت با استفاده از معادله [؟] و [؟] می توان رابطه [؟] را نتیجه گرفت و میزان α را تخمین زد.

$$\alpha = (L + D\lambda)^{-1}c \quad (11-3)$$

۵-۱-۳ خلاصه ی روش

درهم تنیدگی نزدیکترین همسایگی در شکل کلی استخراج بیشتر از دو لایه را تضمین می کند، به عبارتی، آلفاهای تخمین زده شده در هر پیکسل معین بصورت زیر محاسبه می شود.

$$(L + \lambda D)\alpha_1 = \lambda v_1$$

⋮

$$(L + \lambda D)\alpha_n = \lambda v_n$$

با جمع روابط فوق خواهیم داشت:

$$(L + \lambda D) \sum_{i=1}^n \alpha_i = \lambda \sum_{i=1}^n v_i = \lambda m \quad (12-3)$$

$$(L + \lambda D)\alpha = \lambda D\alpha = \lambda m \quad (13-3)$$

معادله ؟؟ میزان هر α در هر لایه را تخمین می زند. در عمل درهم تنیدگی نزدیکترین همسایگی، آلفاهای خروجی تقریباً بین [۰، ۱] هستند. برای بدست آوردن آلفا یک لایه، این لایه به عنوان پیش زمینه در نظر گرفته می شود و مابقی به عنوان پس زمینه. سه لایه $L_1 = (0, 0, 1)$ ، $L_2 = (0, 1, 0)$ ، $L_3 = (1, 0, 0)$ و پیکسل $I = (1/3, 1/3, 1/3)$ را در نظر بگیرید. برای بدست آوردن آلفا از L_1 ، L_2 ، L_3 را به عنوان پیش زمینه F و اجتماع L_2 و L_3 را به عنوان پس زمینه B در نظر بگیرید. طبق معادله

$$\alpha = \frac{(I - B)(F - B)}{\|F - B\|^2} \quad (14-3)$$

، مقدار آلفا برای L_1 برابر است با ۰/۵. به همین صورت مقدار آلفا برای L_2 ، L_3 برابر با ۰/۵ است. در نتیجه مجموعشان برابر با ۱/۵ است. نرمال کردن ممکن است کمک کند اما فاکتورهای نرمالیزه از یک پیکسل به پیکسل دیگر متفاوت است و ب همین دلیل نمی توان ویژگی های پیکسل ها را نرمال کرد [؟].










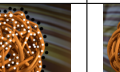





فصل ۴

نتایج و پیشنهادات

۱-۴ نتایج آزمایش

در این فصل آزمایشات مربوط به درهم تیندگی انجام شده است و نتایج آزمایش به صورت مصور و یا عدد نمایش داده شده است. تمامی این آزمایشات با استفاده از نرم افزار برنامه نویسی Matlab انجام شده اند.

در آزمایش اول با انتخاب ۳ عدد تصویر از سایت Alpha Matting [۲] الگوریتم های Bayesian و Closed Form که در فصل اول شرح داده شده اند و الگوریتم KNN Matting مورد آزمایش قرار گرفته اند. برای اینکه بتوان یک مقایسه ایده آل را داشت، از یک نقشه سه گانه مشترک برای هر سه روش استفاده شده است. نتایج آزمایش به همراه نقشه سه گانه مورد نظر در جدول؟؟ قرار داده شده است:

Bayesian	Closed Form	KNN	Trimap	input	Number
					۱
					۲
					۳

جدول ۴-۱: نتایج سه روش درهم تیندگی

در جدول؟؟ تصاویر پیش زمینه شناسایی شده توسط هر کدام از الگوریتم های مذکور نشان داده شده است. جهت اینکه میزان کارایی این الگوریتم ها سنجیده شود از دو معیار زمان و خطای میانگین مربعات (MSE) استفاده شده است.

میزان کارایی این الگوریتم ها برای این دسته از نمونه در جدول؟؟ قرار داده شده است:








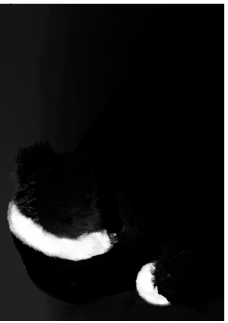
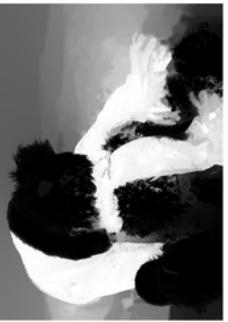



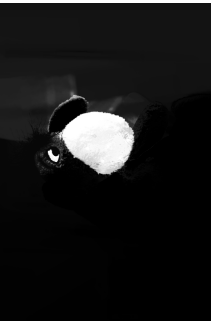
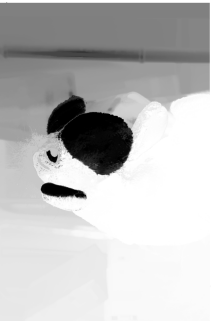

Bayesian		Closed form		KNN		number
MSE	Time	MSE	Time	MSE	Time	n
۶۶۴.۹۸۰۴	۳۴۰.۲۸	۷۹.۳۶۴۸	۱۴.۰۸	۲۸.۲۲۸۱	۲۰.۳۵	۱
۳۲۸.۸۴۷۱	۳۶۸.۹۸	۳۴۹.۸۸۰۸	۱۳.۴۱	۱۹۰.۴۳۶۶	۲۴.۰۰	۲
۱۶۶۹.۴۰۰۴	۴۷۱.۳۴	۲۲۴.۶۲۶۲	۱۵.۲۲۸	۳۸۰.۶۷۵۵	۳۱.۹۰	۳
۶۶۶.۱۰۵۰	۲۸۴.۱۵۴۲	۲۱۷.۹۵۷۲	۱۴.۲۳	۱۹۹.۷۸۰۰	۲۵.۴۱	میانگین

جدول ۴-۲: زمان و میزان خطا برای سه الگوریتم درهم تنیدگی

با توجه به جدول؟؟ رنگ خاکستری کم رنگ برای کمترین میزان خطا و رنگ خاکستری پررنگ برای کمترین زمان در نظر گرفته شده است. می توان به این نتیجه رسید که روش درهم تنیدگی نزدیکترین همسایگی خطای کمتری نسبت به دو روش دیگر دارد و عمل شناسایی پیش زمینه را بهتر می تواند انجام دهد.

کد متلب ارائه شده در سایت Alpha Matting [؟] برای الگوریتم KNN Matting، تنها دو لایه را می تواند از هم جدا کند. ما در این پایان نامه این کد را برای جداسازی چند لایه تعمیم دادیم که نتایج در جدول؟؟ قرار داده شده است. در اینجا برای پیدا کردن هر لایه از یک رنگ مخصوص استفاده می کنیم. همانگونه که در ستون "تصویر نشانه گذاری شده" مشاهده می کنید برای هر لایه تعدادی نشانه با یک رنگ مخصوص بر روی لایه مورد نظر قرار داده شده است. در واقع این نشانه ها نقاطی رنگی هستند. با اینگونه نشانه گذاری می توان هر لایه را به طور جداگانه از پس زمینه جدا کرد. در واقع با این کار می توان اشیاء موجود در یک تصویر را شناسایی کرد. کد مذکور در پیوست قرار دارد.

جدول ۴-۳: نتایج لایه های چندگانه درهم تنیدگی نزدیکترین همسایگی

تصویر ورودی	تصویر نشانه گذاری شده	لایه اول	لایه دوم	لایه سوم
				
				
				

۲-۴ بحث و نتیجه گیری

۱-۲-۴ روش پیشنهادی

در این پایان نامه از روش درهم تنیدگی نزدیکترین همسایگی جهت طبقه بندی داده ها استفاده شده است. بدین صورت که هر بار یک طبقه به عنوان پیش زمینه در نظر گرفته می شود و مابقی طبقات پس زمینه می شوند. سپس داده های مربوط به طبقه پیش زمینه شناسایی می شوند. برای اینکه بتوان این کار را انجام داد نیاز به یک مجموعه داده است. از این مجموعه داده باید تعدادی را به عنوان داده های آموزشی و تعداد دیگر را به عنوان داده آزمایشی در نظر گرفت. برای انتخاب داده های آموزشی در اینجا از روش اعتبار سنجی متقابل^۱ استفاده شده است. به این صورت که هر بار تعدادی از داده ها به عنوان داده آموزشی در نظر گرفته می شود و مابقی به عنوان داده آزمایشی. این عمل برای تمامی داده ها تکرار می شود. عمل درهم تنیدگی بر روی همه این تکرار ها انجام می گیرد و میانگین نتایج به عنوان نتیجه اصلی ارائه می شود. در آزمایشات انجام شده در بخش بعد، مقدار K برای روش اعتبار سنجی متقابل برابر ۳، ۵ و ۱۰ در نظر گرفته شده است.

همانند ساخت نقشه سه گانه در تصاویر، اینجا نیز نیاز به یک نقشه سه گانه است. نشانه گذاری در اینجا به این صورت است که داده های آموزشی پیش زمینه مقدار ۱، داده های آموزشی پس زمینه مقدار ۰ و دیگر داده ها مقدار ۰.۵ می گیرند. برای هر طبقه نقشه سه گانه مربوط به آن طبقه ساخته می شود و عمل درهم تنیدگی نزدیکترین همسایگی روی آن مجموعه داده انجام می شود. مقادیر آلفا مربوط به هر طبقه ذخیره می شود و در نهایت طبق ماتریس آلفا در مورد هر طبقه تصمیم گیری می شود. نکته ای که در اینجا قابل ذکر است این است که در تصاویر مقدار آلفا میزان تیرگی هر پیکسل را نشان می داد و به طور دقیق پیش زمینه یا پس زمینه بودن هر پیکسل را مشخص نمی کنند. اما در اینجا باید به طور دقیق گفته شود که هر داده به کدام طبقه تعلق دارد. برای این کار نیاز به یک سطح آستانه است. شاید تصور بر این باشد که این سطح آستانه ۰.۵ در نظر گرفته شود و بفرض گفته شود مقادیر بیشتر از ۰.۵ مربوط به طبقه پیش زمینه و مقادیر بیشتر از ۰.۵ مربوط به طبقه پس زمینه هستند. اما با آزمایشات انجام شده این نتیجه به دست آمد که سطح آستانه برای هر مجموعه داده و برای هر طبقه متفاوت است و نمی توان به این صورت در نظر گرفته شود. این مقدار آستانه با آزمون و خطا قابل دستیابی است اما این کار بسیار زمان بر است.

ایده ای که در این پایان نامه برای حل این مسئله به کار برده شده است استفاده از روش خوشه بندی Kmeans است. طبقه این الگوریتم خوشه بندی، مقادیر آلفا به تعداد خوشه های مورد نظر تقسیم بندی می شوند. استفاده از این الگوریتم خوشه بندی، نتایج بهتری را نسبت به روش های دیگر طبقه بندی حاصل شده است.

^۱K fold cross validation

۲-۲-۴ آزمایشات

در این پایان نامه روش درهم تنیدگی، برای طبقه بندی داده ها بر روی تعدادی مجموعه داده انجام شده است. اطلاعات مربوط به این مجموعه داده ها در جدول ۴-۲ آمده است. نتایج این آزمایش با سه روش مهم طبقه بندی نزدیکترین همسایگی^۲، درخت تصمیم^۳ [؟] و ماشین بردار پشتیبان^۴ [؟] مقایسه شده است. دو معیاری که در این آزمایشات مورد بررسی قرار گرفته اند، میزان صحت و زمان اجرا است. در هر جدول در هر سطر بهترین نتیجه با زمینه خاکستری رنگ نشان داده شده است. سطر آخر هر جدول نیز میانگین نتایج برای هر ستون را نشان می دهد که در اینجا نیز بهترین مورد با زمینه خاکستری نشان داده شده است.

ردیف	مجموعه داده	شرح	تعداد نمونه	تعداد ویژگی
۱	amlAll	مجموعه داده مربوط به بیماری سرطان لوسمی است که به دو نوع aml، all تقسیم بندی می شوند.	۷۲	۷۱۲۹
۲	fisheriris	مجموعه داده های معروف گل های زنبق	۱۵۰	۴
۳	bankruptcy	داده های مربوط به ورشکستگی بانکها	۶۶	۲
۴	dermathology	هدف از این مجموعه داده ها، تعیین نوع بیماری اریتماتوز اسکواموس است.	۳۶۶	۳۳
۵	ecoli	مجموعه داده موقعیت های اجتماع پروتئین ها	۳۳۲	۶
۶	ionosphere	مجموعه داده طبقه بندی بازگشت رادار از یونوسفر	۳۵۱	۳۴
۷	Sonar	شناسایی معادن و سنگ های معدنی	۲۰۸	۶۰
۸	tworingr	داده های معروف tworing	۲۷۳	۲
۹	twomoon	داده های معروف twomoon	۲۵۸	۲

جدول ۴-۴: اطلاعات مربوط به مجموعه داده های استفاده شده در آزمایشات

نتایج مربوط به آزمایشات را در جداول زیر مشاهده می کنید.

طبق جدول ۴-۲ هنگامی که ۳ داده انتخاب می شود زمان اجرا در الگوریتم نزدیکترین همسایگی کمتر از الگوریتم های دیگر است اما طبق جدول ۴-۲؟ میزان صحت در انتخاب ۳ داده در الگوریتم درهم تنیدگی با نزدیکترین همسایگی بیشتر از دیگر الگوریتم ها است. طبق جداول ۴-۲ و ۴-۲؟ هنگامی که ۵ داده انتخاب شود زمان اجرا برای الگوریتم درخت تصمیم کمترین و میزان صحت در الگوریتم درهم تنیدگی با نزدیکترین همسایگی بیشترین است و در نهایت طبق جدول ۴-۲ و ۴-۲؟ کمترین زمان اجرا مربوط به الگوریتم نزدیکترین همسایگی و بیشترین میزان صحت مربوط به الگوریتم درهم تنیدگی با نزدیکترین همسایگی است.

^۲NN

^۳Decision Tree

^۴SVM

روش				ردیف
KNNMatting	NN	SVM	DT	
۲/۶۴۳	۰/۰۴۱	۰/۱۲۲	۰/۰۷۹	۱
۲/۱۳۷	۰/۰۱۷	۰/۰۵۹	۰/۰۱۱	۲
۲/۱۴۸	۰/۰۱۲	۰/۰۳۲	۰/۰۱۰	۳
۲/۳۰۰	۰/۰۱۳	۰/۰۶۰	۰/۰۱۰	۴
۲/۱۶۰	۰/۰۱۲	۰/۰۴۸	۰/۰۱۰	۵
۲/۱۴۶	۰/۰۱۲	۰/۰۳۲	۰/۰۱۰	۶
۲/۱۰۶	۰/۰۱۲	۰/۰۳۲	۰/۰۱۰	۷
۲/۱۴۳	۰/۰۱۲	۰/۰۳۱	۰/۰۱۰	۸
۲/۱۲۷	۰/۰۱۲	۰/۰۳۱	۰/۰۱۰	۹
۲/۲۱۲	۰/۰۱۶	۰/۰۵۰	۰/۰۱۸	میانگین

جدول ۴-۵: زمان اجرا و تیکه ۳ داده انتخاب می شود

روش				ردیف
KNNMatting	NN	SVM	DT	
۰/۶۴۶	۰/۶۳۹	۰/۶۳۴	۰/۶۳۹	۱
۰/۸۵۷	۰/۳۲۷	۰/۷۵۱	۰/۳۲۷	۲
۰/۸۷۶	۰/۴۹۲	۰/۸۲۳	۰/۴۹۲	۳
۰/۵۲۸	۰/۲۹۴	۰/۵۰۹	۰/۲۹۴	۴
۰/۳۲۳	۰/۴۳۱	۰/۴۶۰	۰/۴۳۱	۵
۰/۶۴۰	۰/۶۱۹	۰/۶۲۶	۰/۶۱۹	۶
۰/۵۱۷	۰/۵۰۴	۰/۵۴۸	۰/۵۰۴	۷
۱/۰۰۰	۰/۵۷۵	۰/۷۹۰	۰/۵۷۵	۸
۱/۰۰۰	۰/۵۹۲	۰/۵۴۹	۰/۵۹۲	۹
۰/۷۱۰	۰/۴۹۷	۰/۶۳۲	۰/۴۹۷	میانگین

جدول ۴-۶: میزان صحت و تیکه ۳ داده انتخاب می شود

روش				ردیف
KNNMatting	NN	SVM	DT	
۲/۵۳۷	۰/۰۲۵	۰/۰۶۸	۰/۰۱۶	۱
۲/۱۳۰	۰/۰۱۲	۰/۰۵۶	۰/۰۱۱	۲
۲/۱۴۰	۰/۰۱۲	۰/۰۳۳	۰/۰۱۰	۳
۲/۲۱۱	۰/۰۱۲	۰/۱۱۶	۰/۰۱۰	۴
۲/۱۵۳	۰/۰۱۲	۰/۰۷۸	۰/۰۱۰	۵
۲/۲۸۱	۰/۰۱۱	۰/۰۳۲	۰/۰۱۰	۶
۲/۱۱۵	۰/۰۱۱	۰/۰۳۲	۰/۰۱۰	۷
۲/۱۴۸	۰/۰۱۲	۰/۰۳۱	۰/۰۱۰	۸
۲/۱۸۱	۰/۰۱۲	۰/۰۳۳	۰/۰۱۲	۹
۲/۲۱۱	۰/۰۱۳	۰/۰۵۳	۰/۰۱۱	میانگین

جدول ۴-۷: زمان اجرا و تیکه ۵ داده انتخاب می شود

روش				ردیف
KNNMatting	NN	SVM	DT	
۰٫۷۰۷	۰٫۶۵۳	۰٫۶۲۳	۰٫۶۵۳	۱
۰٫۸۵۶	۰٫۳۲۹	۰٫۸۳۷	۰٫۳۲۹	۲
۰٫۹۳۳	۰٫۵۲۷	۰٫۸۹۴	۰٫۴۹۲	۳
۰٫۷۱۱	۰٫۳۰۶	۰٫۶۷۹	۰٫۳۰۶	۴
۰٫۴۰۹	۰٫۴۳۱	۰٫۵۸۷	۰٫۴۳۱	۵
۰٫۶۷۵	۰٫۶۴۱	۰٫۶۶۵	۰٫۶۴۱	۶
۰٫۵۳۷	۰٫۵۰۸	۰٫۶۰۶	۰٫۵۰۸	۷
۱٫۰۰۰	۰٫۶۱۲	۰٫۸۳۹	۰٫۶۱۲	۸
۱٫۰۰۰	۰٫۶۲۴	۰٫۵۵۰	۰٫۶۲۴	۹
۰٫۷۵۹	۰٫۵۱۵	۰٫۶۹۸	۰٫۵۱۱	میانگین

جدول ۴-۸: میزان صحت و قتیکه ۵ داده انتخاب می شود

روش				ردیف
KNNMatting	NN	SVM	DT	
۲٫۷۴۵	۰٫۰۳۵	۰٫۰۷۸	۰٫۰۵۰	۱
۲٫۲۰۶	۰٫۰۱۳	۰٫۰۶۲	۰٫۰۱۵	۲
۲٫۱۶۱	۰٫۰۱۲	۰٫۰۳۴	۰٫۰۱۰	۳
۲٫۲۴۷	۰٫۰۱۲	۰٫۱۷۰	۰٫۰۱۰	۴
۲٫۱۶۱	۰٫۰۱۲	۰٫۱۲۳	۰٫۰۱۰	۵
۲٫۱۲۴	۰٫۰۱۲	۰٫۰۳۲	۰٫۰۱۰	۶
۲٫۱۱۳	۰٫۰۱۲	۰٫۰۳۲	۰٫۰۱۰	۷
۲٫۱۳۳	۰٫۰۱۳	۰٫۰۳۲	۰٫۰۰۹	۸
۲٫۱۷۸	۰٫۰۱۲	۰٫۰۳۳	۰٫۰۱۴	۹
۲٫۲۳۰	۰٫۰۱۵	۰٫۰۶۶	۰٫۰۱۵	میانگین

جدول ۴-۹: زمان اجرا و قتیکه ۱۰ داده انتخاب می شود

روش				ردیف
KNNMatting	NN	SVM	DT	
۰٫۷۳۲	۰٫۶۷۶	۰٫۶۸۳	۰٫۵۶۷	۱
۰٫۸۹۰	۰٫۸۶۶	۰٫۹۲۶	۰٫۵۶۰	۲
۰٫۹۳۰	۰٫۹۴۷	۰٫۹۰۴	۰٫۶۸۴	۳
۰٫۷۸۹	۰٫۵۷۱	۰٫۷۹۴	۰٫۴۰۷	۴
۰٫۴۵۰	۰٫۵۶۳	۰٫۶۸۴	۰٫۵۴۸	۵
۰٫۷۰۳	۰٫۶۵۷	۰٫۷۴۰	۰٫۷۰۷	۶
۰٫۵۸۹	۰٫۵۷۱	۰٫۶۵۷	۰٫۶۱۳	۷
۱٫۰۰۰	۰٫۸۶۷	۰٫۸۸۲	۰٫۷۶۰	۸
۱٫۰۰۰	۰٫۶۴۶	۰٫۵۷۱	۰٫۵۶۷	۹
۰٫۷۸۷	۰٫۷۰۷	۰٫۷۶۰	۰٫۶۰۱	میانگین

جدول ۴-۱۰: میزان صحت و قتیکه ۱۰ داده انتخاب می شود

۳-۲-۴ نتیجه گیری

در این تحقیق چند حوزه درهم‌تنیدگی تصویر مورد بررسی و مقایسه قرار گرفتند. در میان روش‌های درهم‌تنیدگی بهترین روش، درهم‌تنیدگی مبتنی بر نزدیکترین همسایگی معرفی شد. از این روش برای طبقه‌بندی داده‌های غیر تصویری استفاده شد. بر اساس اطلاع نگارندگان این اولین بار است که از شیوه‌های درهم‌تنیدگی تصویر برای طبقه‌بندی داده‌های غیر تصویری استفاده می‌شود. این شیوه قابلیت محاسبه چند طبقه (دسته) را دارد. در روش‌های درهم‌تنیدگی تصویر داده‌های آموزشی و آزمایشی در قالب نقشه سه‌گانه به الگوریتم داده می‌شود و میزان تعلق هر پیکسل به هر طبقه توسط الگوریتم تخمین زده می‌شود. در این تحقیق شیوه استفاده از این روش برای داده‌های غیر تصویری بیان شد و این روش روی مجموعه داده‌های مختلف بررسی شد. نتایج آزمایشات انجام شده دقت بالاتر شیوه پیشنهادی را نسبت به برخی از روش‌های مشهور طبقه‌بندی داده‌ها نشان داد. یک مزیت دیگر این روش این است که برای تعداد نمونه‌های کم به خوبی کار می‌کند. یعنی نیازی نیست که تعداد نمونه آموزشی زیادی را در نظر بگیریم بلکه اگر از هر دسته یک داده را داشته باشیم می‌توانیم به بهترین نحو عمل طبقه‌بندی را انجام دهیم. نقطه ضعف فعلی این شیوه در قیاس با دوروش دیگر زمان اجرای کندتر آن است.

۴-۲-۴ کارهای آتی

برای ادامه این کار استفاده از روش خوشه‌بندی gmm^۱ بجای الگوریتم خوشه‌بندی K means و راه حل کاهش زمان اجرا پیشنهاد می‌شود. امید است نتایج بهتری حاصل شود.

^۱Gradient Mixture Model

فهرست منابع

- [۱] محمد، صنیعی آباذه،. داده کاوی کاربردی. انتشارات نیاز دانش، ۱۳۹۳.
- [۲] ، رشید آبادی، فرزانه امین طوسی محمود. برش هوشمند تصویر. پایان نامه کارشناسی ارشد، دانشکده ریاضی و علوم کامپیوتر، دانشگاه حکیم سبزواری، ۱۳۹۴.
- [3] Wang, Jue and Cohen, Michael F. Image and video matting: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(2):97–175, 2008.
- [4] Yuanjie Zheng, Chandra Kambhamettu. Learning based digital matting.
- [5] Ruzon, M. and Tomasi, C. Alpha estimation in natural images. *Proceedings of IEEE CVPR*, pp. 18–25, 2000.
- [6] Y.-Y. Chuang, B. Curless, D. H. Salesin and Szeliski, R. A bayesian approach to digital matting. *Proceedings of IEEE CVPR*, pp. 264–271, 2001.
- [7] Comaniciu, D. and Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
- [8] CORPORATION, C. Knockout user guide. 2002.
- [9] J. Sun, J. Jia, C.-K. Tang and Shum, H.-Y. Poisson matting. *Proceedings of ACM SIGGRAPH*, pp. 315–321, 2004.
- [10] A. Levin, D. Lischinski and Weiss, Y. A closed form solution to natural image matting. *Proceedings of IEEE CVPR*, 30(2):228–242, 2006.
- [11] Lee, P. and Wu, Y. Nonlocal matting. *Conference paper in: CVPR*, 2011.
- [12] Trevor Hastie, Robert Tibshirani and Friedman, Jerome. *The Elements of Statistical Learning*.
- [13] Deza, Elena; Deza, Michel Marie. *Encyclopedia of distances*. 2009.
- [14] Tibco spotfire. <http://www.docs.tibco.com>.
- [15] James M. Abello, Panos M. Pardalos and Resende, Mauricio G. C. *Handbook of massive data sets*. 2002.

- [16] Hamming, Richard W. Error detecting and error correcting codes. *Bell System Technical Journal*, 29:147–160, 1950.
- [17] Jaccard, Paul. The distribution of the flora in the alpine zone. *New Phytologist*, 6(2), 1912.
- [18] Finkel, R.A. and Bentley, J.L. Quad trees - a data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.
- [19] Jerome H. Friedman, Jon Louis Bentley, Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. 3(3):209–226, 1975.
- [20] Blum, M.; Floyd, R. W.; Pratt V. R.; Rivest R. L.; Tarjan R. E. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4), 1973.
- [21] Cormen, Thomas H.; Leiserson, Charles E.; Rivest Ronald L. *Introduction to Algorithms*.
- [22] Chandran, Sharat. Introduction to kd-trees. *University of Maryland Department of Computer Science*, 2004.
- [23] R. Barrett, M. Berry, T. F. Chan J. Demmel J. Donato J. Dongarra V. Eijkhout R. Pozo C. Romine and der Vorst., H. V. Templates for the solution of linear systems: Building blocks for iterative methods, 2nd edition. *SIAM, Philadelphia*, 1994.
- [24] Qifeng Chen, Dingzeyu Li and Tang, Chi-Keung. Knn matting. *IEEE Computer Society*, 35(9):2175–2188, 2013.
- [25] Alpha matting. <http://www.alphamatting.com>.
- [26] S., Murthy. Automatic construction of decision trees from data: A multidisciplinary survey. *Data Mining and Knowledge Discovery*.
- [27] Cristianini, Nello and Shawe-Taylor, John. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

پیوست آ

برنامه های MATLAB روش های مطرح شده در

پایان نامه

۱-آ روش Bayesian

برنامه ۱-آ: کد MATLAB روش Bayesian

```
%Considered FG if above FG_THRESH  
%BG if lower than BG_THRESH  
%Unknown otherwise  
FG_THRESH = 255.0*.95;  
BG_THRESH = 255.0*.05;  
%for each pixel, iterate with max  
%iterations NUM_ITER  
%or if alpha value does not change by at least EPSILON  
NUM_ITER = 10;  
EPSILON = .0001;  
%Read in images  
[FileName,PathName,FilterIndex] = uigetfile({'*.jpg;*.jpeg;*.png;*.gif  
;*.bmp;*.tiff', ...  
'Image Files (.jpg, .png, .gif, .bmp, .tiff}','Select Image To  
Segment');  
img1 = imread([PathName FileName]);
```

```

[FileName,PathName,FilterIndex] = uigetfile({'*.jpg;*.jpeg;*.png;*.gif
      *.bmp;*.tiff', ...
      'Image Files (.jpg, .png, .gif, .bmp, .tiff}','Select Trimap Image'
      );
img2 = imread([PathName FileName]);
img2 = img2(:,:,1); % Added by M.Amintoosi, for 3 layers map images
img1_r = size(img1, 1);
img1_c = size(img1, 2);
channel_sz = img1_r * img1_c;
%Get foreground and background pixels
f_ind = find(img2 >= FG_THRESH); f_ind = [f_ind; f_ind+channel_sz;
      f_ind+channel_sz*2];
b_ind = find(img2 <= BG_THRESH); b_ind = [b_ind; b_ind+channel_sz;
      b_ind+channel_sz*2];
u_ind = find(img2 < FG_THRESH & img2 > BG_THRESH); u_ind = [u_ind;
      u_ind+channel_sz; u_ind+channel_sz*2];
fg = double(img1); fg(b_ind) = 0; fg(u_ind) = 0;
bg = double(img1); bg(f_ind) = 0; bg(u_ind) = 0;
un = double(img1); un(f_ind) = 0; un(b_ind) = 0;
figure('Name', 'FG, BG, UNKNOWN'); imshow([uint8(fg) uint8(bg) uint8(un
      )]); drawnow;
%Compute statistics from known foreground + background
f_bar = [0 0 0];
b_bar = [0 0 0];
%Find mean
for i=1:3
    f = fg(:,:,i);
    b = bg(:,:,i);
    f_bar(i) = mean(f(find(f)));
    b_bar(i) = mean(b(find(b)));
end
%Find covariance
f_tmp(:,:,1) = fg(:,:,1) - f_bar(1);
f_tmp(:,:,2) = fg(:,:,2) - f_bar(2);

```



```

f_tmp(:,:,3) = fg(:,:,3) - f_bar(3);
sumF = [0 0 0; 0 0 0; 0 0 0];
count = 0;
for c=1:size(f_tmp, 2)
    for r=1:size(f_tmp, 1)
        pixF = f_tmp(r,c, :);
        pixF = reshape(pixF,3,1);
        if(any(fg(r,c,:)))
            sumF = sumF + (pixF * pixF');
            count = count + 1;
        end
    end
end
f_co = sumF / count;
%Cov of B
b_tmp(:,:,1) = bg(:,:,1) - b_bar(1);
b_tmp(:,:,2) = bg(:,:,2) - b_bar(2);
b_tmp(:,:,3) = bg(:,:,3) - b_bar(3);
sumB = [0 0 0; 0 0 0; 0 0 0];
count = 0;
for c=1:size(b_tmp, 2)
    for r=1:size(b_tmp, 1)
        pixB = b_tmp(r,c, :);
        pixB = reshape(pixB,3,1);
        if(any(bg(r,c,:)))
            sumB = sumB + (pixB * pixB');
            count = count + 1;
        end
    end
end
b_co = sumB / count;
alpha_un = double(img2)/255.0;
F_un = un;
B_un = un;

```

```

%Solve for unknown pixel values and alpha
for c=1:img1_c
    for r=1:img1_r
        if(~any(un(r,c,:)))
            continue
        end
        un_c = reshape(un(r,c,:), 3,1);
        %init alpha as avg of neighbors
        alpha = 0;
        count = 0;
        try
            alpha = alpha + alpha_un(r-1, c);
            count = count + 1;
        end
        try
            alpha = alpha + alpha_un(r+1, c);
            count = count + 1;
        end
        try
            alpha = alpha + alpha_un(r, c-1);
            count = count + 1;
        end
        try
            alpha = alpha + alpha_un(r, c+1);
            count = count + 1;
        end
        try
            alpha = alpha + alpha_un(r-1, c-1);
            count = count + 1;
        end
        try
            alpha = alpha + alpha_un(r+1, c-2);
            count = count + 1;
        end
    end
end

```

```

    try
alpha = alpha + alpha_un(r+1, c+1);
count = count + 1;
end

    try
alpha = alpha + alpha_un(r-1, c+1);
count = count + 1;
end

alpha = alpha / count;
%iteratively solve for color of pixel
for k=1:NUM_ITER
    alpha_prev = alpha;
    [F B] = solveForFB(f_co, b_co, f_bar, b_bar, un_c, alpha);
    alpha = dot((un_c - B), (F - B)) / norm(F-B).^2;
    if(abs(alpha -alpha_prev) <= EPSILON)
        break;
    end
end

alpha_un(r,c) = alpha;
F_un(r,c,:) = F;
B_un(r,c,:) = B;

end

end

%alpha_un = alpha_un / max(alpha_un(:));
f_ind = find(img2 >= FG_THRESH);
b_ind = find(img2 <= BG_THRESH);
u_ind = find(img2 > BG_THRESH & img2 < FG_THRESH);
alpha_un(b_ind) = 0; alpha_un(f_ind) = 1;
figure('Name', 'Alpha Matte'); imshow(alpha_un);
disp('done finding alpha matte');

```

۲-آ روش Closed Form

برنامه آ-۲: کد MATLAB روش Closed Form

```
function alpha=solveAlpha(I,consts_map ,consts_vals ,varargin)
[h,w,c]=size(I);
img_size=w*h;
A=getLaplacian1(I,consts_map ,varargin {:});
D=spdiags(consts_map (:),0,img_size ,img_size);
lambda =100;
x=(A+lambda*D)\( lambda*consts_map (:).* consts_vals ());
alpha=max(min(reshape(x,h,w) ,1) ,0);
```

۳-آ روش KNN Matting

برنامه آ-۳: کد MATLAB روش KNN Matting

```
function output = knn_matting (input, trimap, lambda, level)
%Modified by Parvin Bakhti
% [10;2] means 10 neighbors with default(level) spatial coherence and
% 2 neighbors with weak spatial coherence.
nn = [10; 10];
[m,n,d] = size(input);
foreground = trimap > 0.99;
background = trimap < 0.01;
all_constraints = foreground + background;
% the first part of the feature vector is the rgb or other color
    information,
% the second part is the spatial factor perturbed by a small amount,
% in the for loop below, the second part will be reduced really
    nonlocally
[a, b] = ind2sub([m n],1:m*n);
```

```

feature_vector = [ reshape(input,m*n,d)';[a;b]/sqrt(m*m+n*n)*level+rand
    (2,m*n)*1e-6];
now=0;
for i=1:size(nn,1)
    kdtree = vl_kdtreebuild(feature_vector);
    ind = vl_kdtreequery(kdtree, feature_vector,feature_vector,'
        NUMNEIGHBORS',nn(i),'MAXNUMCOMPARISONS',nn(i)*3);
    index1 = reshape(repmat(uint32(1:m*n),nn(i),1),[],1);
    index2 = reshape(ind,[],1);
    row(now+1:now+m*n*nn(i),:) = [min(index1, index2), max(index1,
        index2)];
    feature_vector(d+1:d+2,:) = feature_vector(d+1:d+2,+)/100;
    now = now+m*n*nn(i);
end
value = max(1-sum(abs(feature_vector(1:d+2,row(:,1))-feature_vector(1:d
    +2,row(:,2))))/(d+2),0);
A = sparse(double(row(:,1)), double(row(:,2)), value, m*n, m*n);
A = A + A';
D = spdiags(sum(A,2), 0, n*m, n*m);
L = D - A;
H = L +lambda*spdiags(all_constraints, 0, m*n, m*n);
iH = ichol(H);
x = pcg(H, lambda*foreground, [], 2000, iH, iH');
output = reshape(x,m,n);
end

```

برنامه آ-۴: کد MATLAB روش KNN Matting برای لایه های چندگانه

```

lambda = 100;
level = 1;
% [10,2]
nn=[10,2];
I = im2double(imread('retinal1.jpg'));
mI = im2double(imread('retinal1-1.jpg'));
mI2=mI;

```

```

[m,n,d]=size(mI);
consts_map=sum(abs(I-mI),3);
map=consts_map;
%
]V,B] = bwboundaries(map,'noholes');
for i=1:length(V)
    T{i}=find(B==i);
    X(i)=mode(mI2(T{i}));
end
    for i=1:length(V)
        for j=1:length(V)
            if X(i)==X(j) && ( i~=j)
                T{i}=union(T{i},T{j});
                T{j}=[];
            end
        end
    end
    emptyCells = cellfun('isempty', T);
T(all(emptyCells,1)) = [];
%
mI2=mI2(:,:,1);
mI2(:,:,:)=0.5;
mI2(T{1})=1;
mI2(T{2})=0;
mI2(T{3})=0;
mI2(T{4})=0;
mI2 = reshape(mI2(:,:,1), [], 1);
foreground = mI2 > 0.99;
background = mI2 < 0.01;
all_constraints = foreground + background;
%
%
]a, b] = ind2sub([m n],1:m*n);
feature_vector = [ reshape(I,m*n,d)';[a;b]/sqrt(m*m+n*n)*level+rand(2,m

```

```

        *n)*1e-6];
now=0;
for i=1:size(nn,1)
%   KD-Tree
    kdtree = vl_kdtreebuild(feature_vector);
    ind = vl_kdtreequery(kdtree, feature_vector,feature_vector,'
        NUMNEIGHBORS',nn(i),'MAXNUMCOMPARISONS',nn(i)*3);
    index1 = reshape(repmat(uint32(1:m*n),nn(i),1),[],1);
    index2 = reshape(ind,[],1);
    row(now+1:now+m*n*nn(i),:) = [min(index1, index2), max(index1,
        index2)];
    feature_vector(d+1:d+2,:) = feature_vector(d+1:d+2,+)/100;
    now = now+m*n*nn(i);
end
%       1-x
value = max(1-sum(abs(feature_vector(1:d+2,row(:,1))-feature_vector(1:d
    +2,row(:,2))))/(d+2),0);
A = sparse(double(row(:,1)), double(row(:,2)), value, m*n, m*n);
A = A + A';
D = spdiags(sum(A,2), 0, n*m, n*m);
%
L = D - A;
%   H
H = L +lambda*spdiags(all_constraints, 0, m*n, m*n);
iH = ichol(H);
x = pcg(H, lambda*foreground, [], 2000, iH, iH');
output = reshape(x,m,n);
toc
imshow(output)

```

۴-آ عمل طبقه بندی داده ها با درهم تیدگی نزدیکترین همسایگی

برنامه آ-۵: کد استفاده از روش درهم‌تنیدگی نزدیکترین همسایگی جهت عمل طبقه‌بندی داده‌های غیر تصویری

```
function yPredict=KNNMattingforArticle(xTrain,yTrain,xTest,yTest)
%Modified by Mahmood Amintoosi & Parvin Bakhti
disp('starting the KNN Matting demo');
run ('.../v1_setup');
lambda = 100;
level = 1;
nn=[10,2];
X=cat(1,xTrain,xTest);
[r,c]=size(X);
if mod(r,2)==0
    X=reshape(X,r/2,2,c);
elseif mod(r,3)==0
    X=reshape(X,r/3,3,c);
elseif mod(r,5)==0
    X=reshape(X,r/5,5,c);
elseif mod(r,7)==0
    X=reshape(X,r/7,7,c);
end
x2=X;
Xtrain2=xTrain(:,1);
Xtest2=xTest(:,1);
%
B=unique(yTrain);
for w=1:length(B)
    C=find(yTrain==B(w));
%
    Xtest2(:,1)=0.5;
    Xtrain2(:,1)=0;
    Xtrain2(C)=1;
x22= cat(1,Xtrain2,Xtest2);
[m,n,d]=size(x2);
    x22=reshape(x22(:,:,1), [], 1);
    foreground = x22 (:,1)> 0.99;
```



```

background = x22(:,1) < 0.01;
all_constraints = foreground + background;
[a, b] = ind2sub([m n],1:m*n);
%
feature_vector = [reshape(X,m*n,d)';[a;b]/sqrt(m*m+n*n)*level+
    rand(2,m*n)*1e-6];
now=0;
for i=1:size(nn,1)
%
    kdtree = vl_kdtreebuild(feature_vector);
    ind = vl_kdtreequery(kdtree, feature_vector,feature_vector,
        'NUMNEIGHBORS',nn(i));
    index1 = reshape(repmat(uint32(1:m*n),nn(i),1),[],1);
    index2 = reshape(ind,[],1);
    row(now+1:now+m*n*nn(i),:) = [min(index1, index2), max(
        index1, index2)];
    feature_vector(d+1:d+2,:) = feature_vector(d+1:d+2,+)/100;
    now = now+m*n*nn(i);
end
%
    1-x
value = max(abs(1-sum(abs(feature_vector(1:d+2,row(:,1))-
    feature_vector(1:d+2,row(:,2))))/(d+2)),0);
A = sparse(double(row(:,1)), double(row(:,2)), value, m*n, m*n)
;
A = A + A';
D = spdiags(sum(A,2), 0, n*m, n*m);
L = D - A;
H = L +lambda*spdiags(all_constraints, 0, m*n, m*n);
iH = ichol(H);
output {w}= pcg(H, lambda*foreground, [], 2000, iH, iH');
foreground=0;
background =0;
end
%
    Kmeans

```

```

yP=cell2mat(output);
yPredict=xTest(:,1);
yPredict(:,:)=0;
for i=1:length(B)
    yP1=yP(1:length(yTrain),i);
    yP2=yP(length(yTrain)+1:end,i);
    [idx,C]=kmeans(yP1,length(B));
    mx=max(C);
    mn=min(C);
%
    S= find((abs(yP2-max(C))<abs(yP2-min(C))));
    yP2(:,:)=0;
    yP2(S)=B(i);
    yPredict(S)=yP2(S);
end
end

```

واژه‌نامه فارسی به انگلیسی

Threshold	آستانه گیری
Partition	افراز
Cross validation	اعتبار سنجی متقابل
Ill-posed	بد طرح
Ill-conditioned	بد وضع
Overfitting	بیش برازش
Background	پس زمینه
Foreground	پیش زمینه
Labeled pixels	پیکسل های برچسب گذاری شده
Unlabeled pixels	پیکسل های برچسب گذاری نشده
Neighboring pixels	پیکسل های مجاور
Prediction function	تابع پیش بینی
Objective function	تابع هدف
Cost function	تابع هزینه
Convert	تبدیل کردن
Extracting	تخمین
Convexity	تحدب
Transpose	ترانهاده
Compositing	ترکیب
Approximation	تقریب
Propagation	تعمیم
Constant	ثابت
Substituting	جای گذاری

Inner producte	حاصل ضرب داخلی
Eliminated	حذف کردن
Scribble	نشان گذاری
Coefficients	ضرایب
Training data	داده های آموزشی
Interpolates	درونیابی
Image Matting	در هم تنیدگی تصویر
Dual	دوگان
Linear regression	رگرسیون خطی
Ridge regression	رگرسیون ریج
Gray level	سطح خاکستری
Pseudo-inverse	شبه معکوس
Intensities	شدت روشنایی
Assumption	فرض
Image segmentation	قطعه بندی تصویر
Least squares errors	کم ترین مربعات خطا
Covariance	کواریانس
Minimizing	کمینه سازی
Diagonal matrix	ماتریس قطری
Training set	مجموعه آموزشی
Data set	مجموعه داده ها
Unknown	مجهول
Evaluation	محاسبه
Linear model	مدل خطی
Local	محلی
Compar	مقایسه
Independent	مستقل
Normal equation	معادلات نرمال
Criterion	معیار
Regularisation	منظم سازی

Invertible	معکوس پذیر
Non smooth	ناهموار
Region	ناحیه
Consequence	نتیجه
Feature mapping	نگاشت طرح
Trimap	نقشه سه گانه
Positive semidefinite	نیمه معین مثبت
Representation	نمایش
Semi-supervised	نیمه نظارتی
Goal	هدف
Overlap	هم پوشانی
Machine learning	یادگیری ماشین
Variance	واریانس

واژه‌نامه انگلیسی به فارسی

Approximation	تقریب
Assumption	فرض
Background	پس زمینه
Coefficients	ضرایب
Compar	مقایسه
Compositing	ترکیب
Constant	ثابت
Convert	تبدیل کردن
Convexity	تحدب
Cost function	تابع هزینه
Consequence	نتیجه
Covariance	کوارینانس
Criterion	معیار
Cross validation	اعتبار سنجی متقابل
Data set	مجموعه داده ها
Diagonal matrix	ماتریس قطری
Dual	دوگان
Eliminated	حذف کردن
Evaluation	محاسبه
Extracting	تخمین
Feature mapping	نگاشت طرح
Foreground	پیش زمینه
Goal	هدف

Gray level	سطح خاکستری
Ill-conditioned	بد وضع
Ill-posed	بد طرح
Image Matting	در هم تنیدگی تصویر
Image segmentation	قطعه بندی تصویر
Independent	مستقل
Inner product	حاصل ضرب داخلی
Intensities	شدت روشنایی
Interpolates	درونیابی
Invertible	معکوس پذیر
Labeled pixels	پیکسل های برچسب گذاری شده
Least squares errors	کم ترین مربعات خطا
Linear model	مدل خطی
Linear regression	رگرسیون خطی
Local	محلی
Machine learning	یادگیری ماشین
Minimizing	کمینه سازی
Neighboring pixels	پیکسل های مجاور
Non smooth	ناهموار
Normal equation	معادلات نرمال
Objective function	تابع هدف
Overfitting	بیش برازش
Overlap	هم پوشانی
Partition	افراز
Positive semidefinite	نیمه معین مثبت
Prediction function	تابع پیش بینی
Propagation	تعمیم
Pseudo-inverse	شبه معکوس
Region	ناحیه
Regularisation	منظم سازی

Representation	نمایش
Ridge regression	رگرسیون ریج
Semi-supervised	نیمه نظارتی
Scribble	نشان گذاری
Substituting	جای گذاری
Threshold	آستانه گیری
Training data	داده های آموزشی
Training set	مجموعه آموزشی
Transpose	ترانهاده
Trimap	نقشه سه گانه
Unknown	مجهول
Unlabeled pixels	پیکسل های برچسب گذاری نشده
Variance	واریانس

Hakim Sabzevari University

An Outline of MSc. Thesis



دانشگاه حکیم سبزواری

Surname: Bakhti

Name: Parvin

Student No.: ۹۴۱۳۱۳۷۰۱۷

Supervisors: Dr. Mahmood Amintoosi and Dr. Mina Malekzadeh

Faculty of Mathematics and Computer Science

Program: Decision Science and Knowledge Engineering

Title of thesis: Image Segmentation with Image Matting Methods

Keywords: Image matting, semi-supervised learning, segmentation, KNN

Abstract: Segmentation of the foreground and background is an important issue of the semi-supervised learning classification that it is associated with the problem of image processing. Semi-supervised learning means that the problem is modeled based on the user input limits. One of the relatively new ways in this field is so-called image matting technique, which has focused on several studies. In this research, we try to describe and interpret the Rozun and Thomasi, Bayesian matting, Mishima, Knockout, Histogram, Global Color Model, Poisson, and Closed Form method. The basis of this research is the description and use of the KNN Matting method as a more desirable method than the methods described in terms of runtime and accuracy, which results from the most common Closed-Form and the Bayesian Matting have been compared with the KNN Matting. In this research, the KNN Matting method is used to identify the objects in the image. Finally, the KNN Matting method is used to classify non-visual data and the desired results are obtained. In the Image Matting method, the decision is not definitive, and it is possible with this probability. But in the case of data matting, this must be done definitively, which itself causes problems that have been solved in this research.



Hakim Sabzevari University

Faculty of Mathematics and Computer Science

**A Thesis Submitted in Partial Fulfilment of the Requirement for the
Degree of Master of Science in Decision Science and Knowledge
Engineering**

Image Segmentation with Image Matting Methods

Supervisors:

Dr. Mahmood Amintoosi and Dr. Mina Malekzadeh

By:

Parvin Bakhti

January 2018