

توضیحات تمرین سری سوم

سید امیر محمد سادات شکوهی

شماره دانشجویی: 98100175

- همانطور که مشخص است این نقاط در مختصات دکارتی تشکیل دو دایره هم مرکز می دهند. در روش k-means برای سگمنت کردن این نقاط به دو کلاستر در هر مرحله k-means عمود منصف بین دو مرکز کلاستر را می کشد و نقاط هر سمت این عمود منصف را به یک کلاستر جدید نسبت می دهد. پس در نهایت نتیجه این کار این است که نقاطی که با یک خط جدا شده اند به عنوان دو کلاستر هستند. و این امر کاملاً واضح است که با یک خط نمی توان نقاط دو دایره هم مرکز را از هم جدا کرد. من در فایل res02.jpg نتیجه این الگوریتم با دو حالت از پارامترهای اولیه قرار داده ام که در کل نتیجه مناسبی نیستند. برای رفع این مشکل باید نقاط را در دستگاه مختصات قطبی نگاه کرد. در این دستگاه هر دایره تبدیل به یک خط می شود و حالا می توان خیلی راحت این دو خط را با خط دیگری جدا کرد. و نتیجه حاصل از k-means را به مختصات دکارتی بر می گردانیم و می بینیم که نقاط دو دایره به خوبی جدا شده اند که نتیجه بهتری است.
- در این سوال به خاطر اندازه بزرگ عکس و کندی روش mean shift ، اول عکس را تا 8 برابر کوچک می کردم. پس از این هر پیکسل عکس را به برداری 5 بعدی شامل RGBXY تبدیل کردم و همه مولفه ها را به بازه (0,1) اسکیل کردم. تابع mean shift را بر روی این بردار ها اعمال کردم و برای هر پیکسل یک لیبل که مربوط به چه کلاستری است را به شکل یک عکس در نظر گرفتم و این عکس را تا 8 برابر بزرگ کردم. این روش مقداری نویز داشت برای همین بر روی عکس بزرگ شده لیبل ها یک فیلتر مدین اعمال کردم تا مقداری از این نویز کم شود. در آخر دیدم که نتیجه حاصل به اندازه کافی خوب نیست و اشیا با یک رنگ به چند تیکه تبدیل شدند که این به خاطر تاثیر زیاد از حد مولفه xy بوده است. برای حل این مشکل من پس از اسکیل کردن به بازه (0,1) این دو مولفه را در مقدار 0.3 ضرب کردم تا تاثیر مکان پیکسل ها کمتر شود.
- اول با مقداری محاسبات تعداد خوشه های لازم که نزدیک k پیدا کردم و پس از مقدار گرادیان هر نقطه را محاسبه کردم. برای مقدار محاسبه گرادیان فیلتر سوبل را در دو جهت افقی و عمودی اعمال کردم و magnitude این دو مقدار را گرادیان در نظر گرفتم. پس از آن عکس را به فضای LAB بردم و یک برداری ویژگی 5 بعدی برای هر پیکسل شامل LABXY درست کردم. پس از آن در مرحله اول به هر نقطه نزدیک ترین مرکز را نسبت دادم و برای هر مرکز در یک همسایگی 5 در 5 خانه ای با کمترین مقدار گرادیان را انتخاب کردم و مرکز ها را آپدیت کردم. پس از این الگوریتمی که در ادامه توضیح می دهم را ده بار تکرار کردم. در هر مرحله به ازای هر مرکز یک پنجره 2S در 2S را در نظر می گیرم و فاصله هر نقطه تا مرکز را به روش گفته شده در فایل توضیحات تمرین حساب کردم و برای نقاطی که این مقدار از فاصله ای که به مرکز دیگری نسبت داده شده اند کمتر بوده است مرکز این نقاط را آپدیت کردم و فاصله هر نقطه تا مرکز را برای مقایسه با دیگر مراکز نگه

داشتم. در آخر بعد از اینکه همه نقاط یک باز به یک مرکزی نسبت داده شدند مراکز را آپدیت کردم. به طوری که هر مرکز را میانگین مختصات همه نقاط نسبت داده شده به آن مرکز قرار دادم. در آخر ماتریس اینکه هر پیکسل به کدام مرکز نسبت داده شده است را یک عکس در نظر گرفتم و برای یک دست شدن سگمنت ها به تعداد 20 بار فیلتر مدین را اعمال کردم بر روی این ماتریس. سپس مانند روش عنوان شده در اول این سوال گرادیان هر نقطه از این ماتریس را محاسبه کردم. نقاطی که مقدار گرادیان مثبتی داشتند را به عنوان مرز کلاستر ها در نظر گرفتم و این نقاط در عکس را سیاه کردم. به دلیل اسکیل نکردن مولفه های مختلف LABXY مقدار الفا برای k های مختلف متفاوت است به طوری که برای 64 قسمت آلفا برابر 0.5 ، برای 256 قسمت آلفا برابر 1 ، برای 1024 قسمت آلفا برابر 2 و برای 2048 تیکه برابر با 4.8 است.

4. در این سوال به خاطر اینکه رنگ پس زمینه و پرده ها تا مقدار خوبی به هم مشابه بود بنده تصمیم گرفتم تا با استفاده از تمپلت مچینگ در اول کاربر بر روی یک پرده کلیک کند و پس از آن با استفاده از تمپلت مچینگ دیگر پرده ها را پیدا کنم. متاسفانه به دلیل اینکه شکل پرده ها زیاد به هم دیگر شبیه نبود یا شاید بنده خوب پیاده سازی نکردم اما نتیجه مناسبی از این ایده نگرفتم و در آخر این بخش را از کردم حذف کردم. متاسفانه به خاطر کمبود وقت نتوانستم روش دیگری را پیاده و آزمایش کنم و در حال حاضر کد بنده فقط با استفاده از توابع آماده روش felzenszwalb یک سگمنتیشن برای عکس پیدا می کند که به تقریب هر پرده در یک سگمنت است. بنده برای اینکه نتیجه ای از این تمرین بگیرم کاری کردم که نتیجه سگمنت شده به کاربر نشان داده شود و کاربر با انتخاب هر سگمنت این سگمنت به نتیجه نهایی اضافه شود. بنده اطلاع داشتم که شما در کلاس گفتید که این روش به تنهایی قابل قبول نیست ولی بنده نتوانستم روش مناسب تری پیدا کنم.

5. در این سوال اول بنده عکس را به کاربر نشان می دهم و کاربر باید یک منحنی به دور تسبیح بکشد. کشیدن منحنی به این شکل است که از زمانی دکمه سمت چپ موس پایین می رود شروع به ذخیره کردن مختصات موس می کند و در این وضعیت باید یک منحنی به دور تسبیح بکشید. و پس پایان این کار دستتان را از روی دکمه سمت چپ موس بردارید تا پنجره اتومات بسته شود و منحنی مورد نظر به دست آید. پس از آن بر روی عکس پردازش های اولیه انجام می شود و اول با فیلتر گاوسی مقدار نویز ها کم می شود و پس از آن با استفاده از روش canny گرادیان عکس را بدست می آوریم. در ادامه شروع به آپدیت کردن منحنی با روش برنامه نویسی پویا کردم و در هر مرحله نقاط مورد نظر را بر روی عکس کشیدم و به عنوان یک فریم فیلم نهایی اضافه کردم تا زمانی که دیگر نقاط هیچ تکونی نخورند. مشکل اصلی ای که بنده داشتم این بود که تسبیح داخل عکس دارای فرو رفتگی بود و شکل محدبی نداشت و منحنی به صورت یک پوش محدب به دور تسبیح در می آمد. بنده برای حل این مشکل تابع انرژی را مقداری تغییر دادم یک جمله فاصله تا مرکز منحنی هم اضافه کردم. این کار باعث می شد تا نقاط به سمت مرکز کشیده شوند و وارد فرو رفتگی های تسبیح شوند. اما مشکل جدیدی که بوجود می آمد بر اثر اضافه شدن این جمله این بود که در بخش هایی از منحنی نقاط از تسبیح عبور کرده و بیشتر به سمت مرکز کشیده می شدند. برای حل این مشکل بنده جمله فاصله تا مرکز را به مقدار گرادیان تقسیم کردم تا تاثیر این جمله در زمانی که در یک نقطه با گرادیان بالا هستیم کم شود. در انتها تابع انرژی برای نقطه (x_i, y_i) به شکل زیر در آمد:

$$E = (1+it/20)(dis - d)^2 - \mu * \text{gadian}(x_i, y_i) + \beta * ((x_i - cx)^2 + (y_i - cy)^2) / (\text{gadian}(x_i, y_i) * 100 + 1)$$

که در این معادله dis فاصله تا نقطه قبلی در منحنی، d فاصله متوسط بین هر دو نقطه، cx و cy مختصات مرکز منحنی، it شماره ایتريشن است. مقادير بتا و میو را به صورت تجربی به دست آوردم و به ترتیب برابر با 500 و 5000 است. در آخر اینکه بنده در 50 ایتريشن اول فاصله متوسط بین دو نقطه را 10 درصد کم کردم و پس از آن ثابت نگه داشتم.