

۱- طراحی سیستم ردیاب به روش Inside Looking-out (مهلت تحویل: ۱۵ روز پس از پایان

تدریس بخش ۴)

سیستم‌های ردیاب مبتنی بر دوربین از دو تکنیک مختلف استفا می‌کنند. تکنیک اول تحت عنوان (OLI) Outside Looking-in شناخته می‌شود که در آن به کمک دوربین‌های ثابت، مکان و وضعیت (Pos) شیء مورد ردیابی تعیین و دنبال می‌شود. دسته دوم تکنیک‌هایی تحت عنوان (ILO) Outside Looking-in نامگذاری شده‌اند که در آن دوربین و یا دوربین‌ها به جسم تحت ردیابی متصل و بر اساس تحلیل تصاویر آن مکان و وضعیت شیء مورد ردیابی تعیین و دنبال می‌شود. در پروژه جاری مایل به طراحی یک ردیاب ILO هستیم. سیستم‌های ILO یا مبتنی بر تک دوربین و یا مبتنی بر زوج دوربین استریو طراحی می‌شوند. سیستم‌های تک دوربین نیاز به نصب یکسری مارکرهای قابل تشخیص از یکدیگر در محیطی است که ردیابی در آن انجام می‌شود و ضمناً مکان این مارکرها در دستگاه مختصات مشخصی بایستی معلوم باشند. یکی از انواع مارکرهای پرکاربرد برای این منظور بارکدهای دوبعدی است. اما در سیستم‌های دو دوربین این امکان وجود دارد که مساله را بتوان با محدودیتهایی در حال بدون مارکر حل کرد.

در پروژه حاضر بنا داریم یک سیستم ILO مبتنی بر تک دوربین طراحی کنیم. کارهایی که انجام خواهید داد:

۱- مقایسه ردیاب‌های OLI و ILO

۲- طراحی ردیاب ILO تک دوربین و با استفاده از دوربین موبایل

- a. تعدادی بارکد دو بعدی با کدهای متمایز را پرینت بگیرید
- b. یکی از دیوارهای اتاق خود را برای نصب مارکرها کاندید کنید و دستگاه مختصات اتاق را چنان در نظر بگیرید که در آن دیوار اتاق شما صفحه XY باشد. در حل این مساله تماماً از دستگاه مختصات دست چپ استفاده کنید. حال مارکرها را در نقاطی با (X,Y) مشخص نصب کنید (مرکز بارکد در این نقطه قرار گیرد). به این ترتیب تمام مارکرها دارای $Z=0$ هستند.
- c. دوربین موبایل خود را روی وضعیتی قرار دهید که همه پارامترهای آن بصورت دستی تنظیم شده و هیچ یک از پارامترها (شامل exposure time, فوکوس, ISO و wight balance (WB) در وضعیت Auto نباشند و برای داشتن تصویر مناسب آنرا بصورت دستی تنظیم کنید.
- d. عملیات آفلاین: دوربین را کالیبره کنید. برای این منظور بایستی از یک چارت شطرنجی استفاده کنید. برای کالیبراسیون می‌توانید در محیط Matlab از APPS-Image Processing and Computer Vision- Camera Calibration استفاده کنید. خروجی این کد پارامترهای داخلی دوربین، شامل فاصله کانونی،

Principle Point، و پارامترهای اعوجاج و البته پارامترهای خارجی است که آن استفاده نخواهیم کرد. پارامترهای داخلی را برای استفاده آتی ذخیره کنید.

e. عملیات آنلاین:

i. از دوربین تصویر دریافت کنید.

ii. به کمک پارامترهای اعوجاج دوربین، تصویر را اصلاح هندسی کرده تا اعوجاج آن برطرف شود.

iii. مارکرهای موجود در تصویر را آشکار و شناسایی کنید.

iv. با داشتن حداقل سه مارکر غیرهم خط (که بر روی یک خط قرار نگرفته‌اند) وضعیت دوربین شامل

موقعیت و زوایای آنرا با شش درجه آزادی بدست آورید. دقت کنید، با توجه به اینکه مارکرها هر کدام

کد منحصر به فرد خود را دارند، و از طرفی هنگام نصب می‌دانید که هر کدی در چه مختصاتی نصب

شده است، حل مساله سر راست خواهد بود.

f. یک سناریوی تست طراحی کنید و بر اساس آن دقت سیستم ردیاب خود را تخمین بزنید.

۲- کالیبراسیون صفحه نمایش مبتنی بر ویدیوپرژکتور (مهلت تحویل: ۱۴ روز پس از پایان مهلت تکلیف

شماره یک)

پروژه انجام شده توسط آقای Johnny Lee برای کالیبراسیون ویدیوپرژکتور در نمایش تصویر بر روی یک اسکرین را دیدید. مایلیم همین کار را تکرار کنیم با این تفاوت که بجای سنسورهای الکترونیکی قرار گرفته در گوشه‌های اسکرین، از یک دوربین استفاده شود. در انجام این پروژه از لپ‌تاپ برای تولید تصویر استفاده کنید و همزمان از دوربین آن برای تصویربرداری بهره ببرید. همچنین برای تست می‌توانید از ویدیوپرژکتور یکی از کلاس‌ها استفاده کنید.

در انجام پروژه گام‌های زیر را بردارید:

۱- بر روی یک کاغذ سفید A4 یک مستطیل 15×20 سانتی‌متر رسم کنید. مستطیل رسم شده چنان باشد که این کادر در تصویر دوربین لپ‌تاپ براحتی قابل مشاهده باشد. این کاغذ را بر روی یک شیت سفت (مثلا یک تکه کارتون) بچسبانید. این مجموعه اسکرین شما را تشکیل داده و بناست در ادامه تلاش کنید تا در موقعیت‌های مختلف اسکرین، تصویر خودتان را به کمک ویدیوپرژکتور داخل این کادر نمایش دهید.

۲- اسکرین را در محلی قرار دهید که در میدان نمایش ویدیوپرژکتور باشد.

۳- لپ‌تاپ خود را در وضعیتی قرار دهید که به کمک دوربین آن اسکرین قابل مشاهده باشد.

a. تصویر از صحنه تهیه کنید

b. در تصویر حاصل، اسکرین را آشکار و در آن محل چهارگوشه کادر را پیدا کنید (اگر فکر می‌کنید که مثلا با

چسباندن تعدادی دایره رنگی در چهار گوشه کادر، راحتتر می‌توانید کادر تصویر را پیدا کنید، این کار را بکنید)

۴- با استفاده از کد گری، نقاط متناظر در تصویر ویدیو پرژکتور، متناظر با چهارگوشه کادر تصویر را پیدا کنید.

۵- بر روی تصویر خود اصلاح هندسی چنان انجام دهید که با دادن تصوراتان به ویدیوپرژکتور، دقیقا داخل کادر قرار گیرد.

۶- الگوریتم خود را در چند وضعیت مختلف اسکرین تکرار و نتایج آنرا گزارش کنید.

۳- عملگرهای مورفولوژیک – طراحی یک الگوریتم ساده OCR^۱ (مهلت تحویل: ۱۳ آذر)

مایلیم برای شناسایی ارقام تاپپی انگلیسی (قلم Calibri) و فارسی (قلم Titr) یک سیستم شناسایی ساده مبتنی بر عملگرهای مورفولوژیک طراحی کنیم. برای این منظور الگوریتم را بگونه‌ای طراحی می‌کنیم که در یک درخت تصمیم، شناسایی ارقام انجام شود. به عنوان مثال، برای ارقام انگلیسی ممکن است ابتدا بررسی کنیم که آیا رقم مورد نظر دارای حفره هست یا نه و در این صورت ارقام به دو زیرمجموعه {1, 2, 3, 5, 7} و {4, 6, 8, 9, 0} تقسیم می‌شوند. در گامهای بعدی بر اساس دیگر معیاری شکلی می‌توانید هر مجموعه را به زیرمجموعه‌های کوچکتر افراز کنید، بطوریکه در انتها، هر زیرمجموعه تنها شامل یک عضو باشد. در این صورت شناسایی رقم انجام شده است.

الگوریتم خود را در حضور مقادیر مختلف نویز نمک و لفل با احتمال 0, 0.05, 0.1, 0.2 و 0.3 برای نویز بودن هر پیکسل آزمایش کنید.

۴- عملگرهای مورفولوژیک – پیش‌پردازش (مهلت تحویل: ۲۰ آذر)

الگوریتم Rtinal-VesselNet^۲، ابزاری برای شناسایی عروق (رگ) تصاویر شبکه‌ای است.

۱- این ابزار را برای شناسایی عروق بر روی بانک تصویر DRIVE و STARE راه‌اندازی و آزمایش کنید. در هر دو این بانکهای تصویر، عروق توسط کاربر انسانی نیز آشکار و در قالب فایل‌های باینری ارائه شده‌اند.

۲- برای هر بانک بصورت جدا متوسط نسبت پیکسل‌هایی که به درستی به عنوان رگ تشخیص داده‌اید را به تعداد پیکسل‌های تشکیل‌دهنده عروق محاسبه کنید. به این پارامتر Sensitivity گفته می‌شود.

۳- برای هر بانک بصورت جدا متوسط نسبت پیکسل‌هایی که به درستی به عنوان غیررگ تشخیص داده‌اید را به تعداد پیکسل‌های غیررگ محاسبه کنید. به این پارامتر Specificity گفته می‌شود.

۴- با مراجعه به مقالات این حوزه دو الگوریتم مورفولوژیک مختلف به عنوان پیش‌پردازش را انتخاب، پیاده‌سازی و اثر آن را بر دوپارامتر Sensitivity و Specificity بررسی کنید.

¹ Optical Caracter Recognition

^۲ <https://github.com/DeepTrial/Retina-VesselNet>