



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

درس بینایی کامپیوتر

استاد درس جناب آقای دکتر صفابخش

(تمرین سری هفتم)

محسن عبادپور | ۴۰۰۱۳۱۰۸۰ | m.ebadpour@aut.ac.ir

نیمسال اول سال تحصیلی ۱۴۰۱-۱۴۰۲



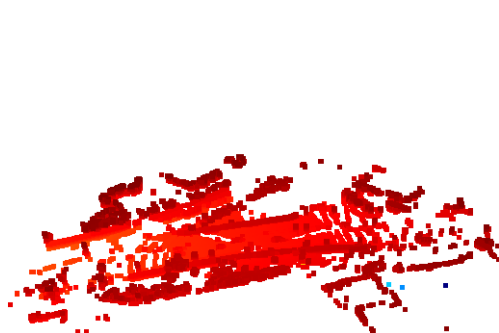
فهرست پاسخ ها

- الف: نمایش ابر نقاط و بیان معایب و مزایا.....۳
- ب: نگاشت ابر نقاط به تصویر.....۵
- ج: درون یابی نقشه عمق.....۸
- د: تشخیص عمق با روش استریو با دو تصویر و مقایسه آن ابر نقاط.....۱۲
- ه: نمایش سه بعدی صحنه.....۱۶

مسئله اول

الف: نمایش ابر نقاط و بیان معایب و مزایا

برای نمایش ابر نقاط، فایل با پسوند bin. مربوط به داده های سنسور لایدار را توسط کتابخانه numpy بصورت باینری خوانده و توسط کتابخانه open3d و کلاس visualization با تابع draw_geometries نمایش می دهیم که بصورت زیر قابل مشاهده می باشد؛ ضمناً خروجی pcb. نیز پیوست شده و با نرم افزار های مصورساز سه بعدی قابل نمایش می باشد.



مهمترین مزیت و نکته‌ی مثبت استفاده از سنسورهای لایدار دقت بسیار بالای آن در اندازه‌گیری عمق و فاصله می‌باشد که باعث می‌شود در کاربرد های حساس و عملیاتی مهم صنعتی مورد استفاده قرار گیرد. مزیت بعدی شیوه نمایش و کار با عمق های تخمینی می باشد بطوری که بتوان فاصله نقاط مختلف را در کنار هم ارزیابی نموده و مقایسه کرد. اما مشکلی بزرگی که داده های حاصل از سنسورهای لایدار دارند این است که عمق تخمینی برای نقاط پیوسته نبوده و حالت شبه sparse می باشد و این باعث می شود ما بصورت پراکنده عمق را تخمین بزنیم (یا حتی بعضی نواحی را کلاً نتوانیم تخمین زنیم) که این مورد چالش های متعددی را به همراه دارد که میتوان به بحث درونی یابی بین نقطه ای، عدم امکان بازشناسی کامل و دقیق سه بعدی صحنه و.. اشاره نمود.

مشکل بعدی استفاده از سنسور های لایدار برای تخمین عمق، محدودیت در تخمین عمق های بسیار مختلف و متفاوت در یک صحنه می باشد بطوری که صرفاً برای پیرامون نزدیک خود میتواند با دقت بسیار خوبی عمق را تخمین زند و برای تخمین عمق محیط دور تر خوب عمل نمی کند (سیگنالی منعکس نمی شود). برای مثال اگر سنسور لایدار را در بالای یک خودرو قرار دهیم و خودرو در وسط یک چهارراه باشد، داده های حاصل از لایدار با دقت خوبی میتواند عمق و فاصله را از خودرو های بغلی خود تخمین بزند ولی برای تخمین فاصله چراغ راهنمایی یا ساختمان های موجود در پیاده رو نمی تواند خوب عمل کند. چالش بعدی استفاده از سنسور های لایدار در تطبیق نقاط تصویر RGB با ابر نقاط می باشد که باعث می شود نتوان با دقت ۱۰۰٪ تعیین نمود که عمق اندازه گیری شده مربوط به کدام بخش از صحنه می باشد لذا توصیف و تفسیر پذیری سه بعدی صحنه نیز با چالش مواجه می شود.

مسئله دوم

ب: نگاشت ابر نقاط به تصویر

روند کلی نگاشت ابر نقاط به تصویر در جلسه آنلاین برگزار شده مورد بررسی قرار گرفت که می توان بصورت خلاصه بدین گونه توضیح داده که ما برای نگاشت ابر نقاط به تصویر ابتدا بایستی مختصات سنسور لایدار را به مختصات دوربین map کنیم و سپس نقاط موجود در خروجی لایدار را به نقاط موجود در دوربین شماره صفر نگاشت کنیم و در گام پایانی نقاط از دوربین شماره صفر به دوربین شماره دو انتقال داده شود. (دوربین شماره دو دوربینی است که ما در حال کار روی تصاویر رنگی آن می باشیم) به عبارتی دیگر فرآیند نگاشت ابر نقاط به تصویر شبیه به تطبیق نقاط دو دوربین به یک دیگر می باشد که در آن ما توسط ماتریس های ذاتی دوربین انتقال را انجام می دهیم.

برای پیاده سازی این بخش، پنج تابع با عناوین زیر پیاده سازی شده است که در مقابل هر یک بصورت مختصر هدف آن آورده شده است:

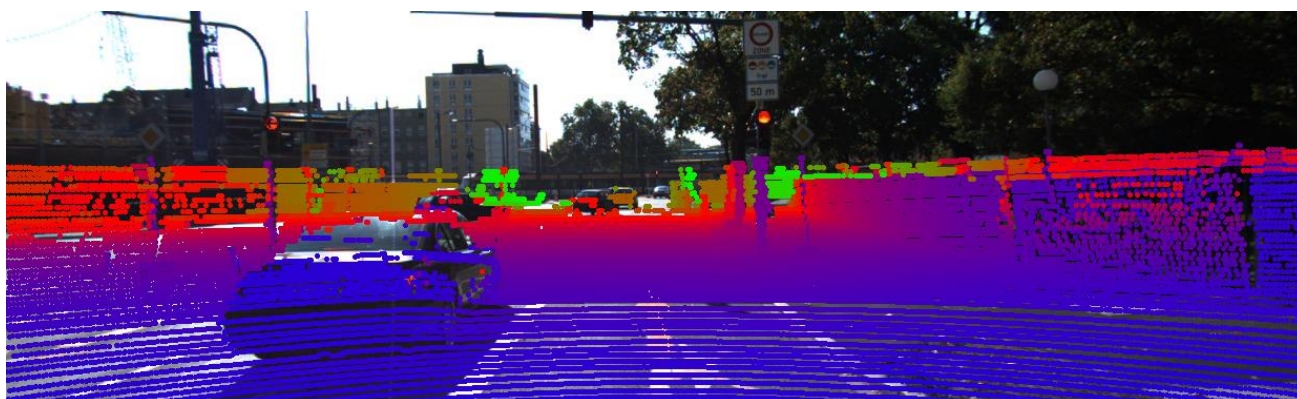
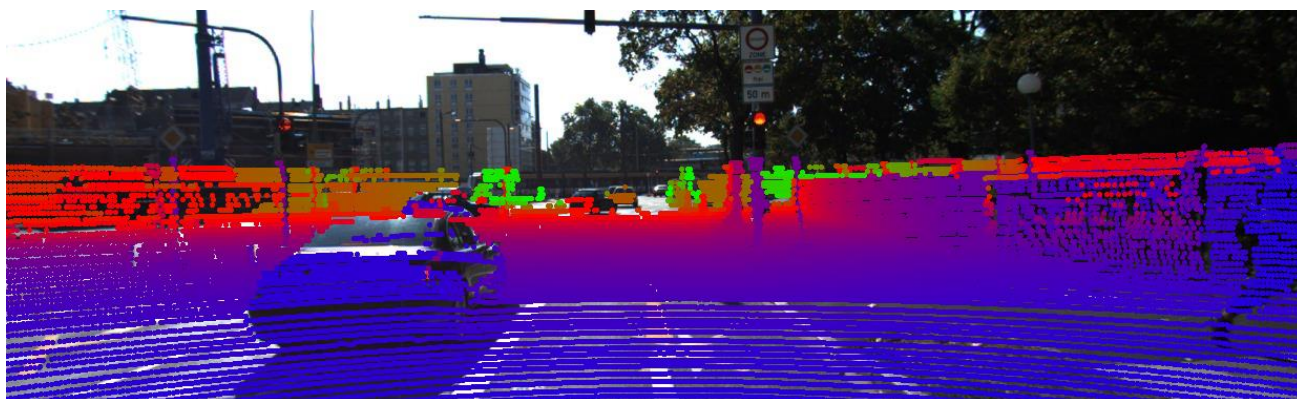
- `read_calib_file`: پارامترهای کالیبراسیون دوربین را از فایل خوانده و ماتریس مربوطه را تشکیل می دهد.
- `read_velo2cam_file`: پارامترهای انتقال و چرخش بین دوربین را از فایل خوانده و ماتریس مربوطه را می سازد.
- `create_projection_matrix`: توسط خروجی دو تابع فوق و ضرب ماتریس های تبدیل، ماتریس نگاشت (projection) را محاسبه می کند.
- `lidar_to_image`: ماتریس نگاشت و ابر نقاط را ورودی گرفته و عمق تخمینی را با مختصات نگاشت شده به تصویر محاسبه و باز میگرداند.

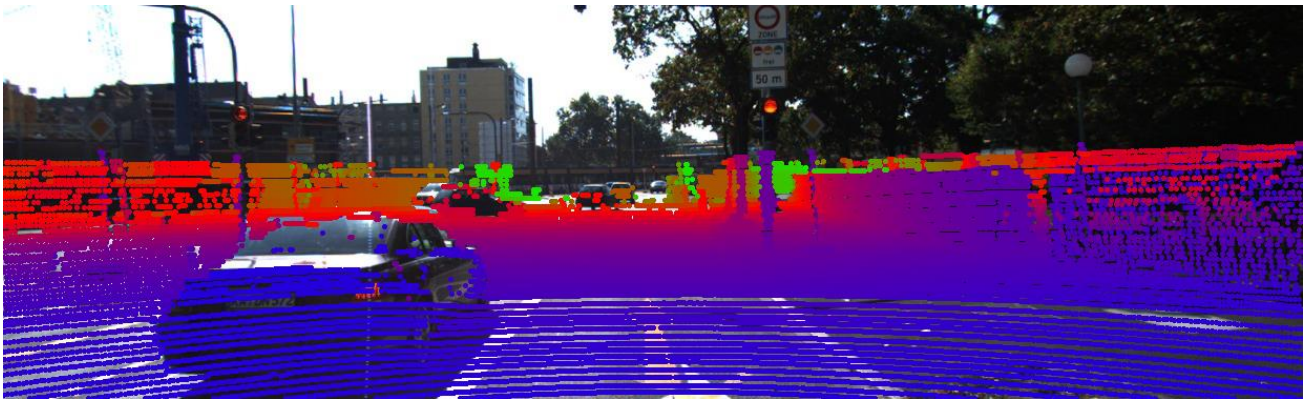
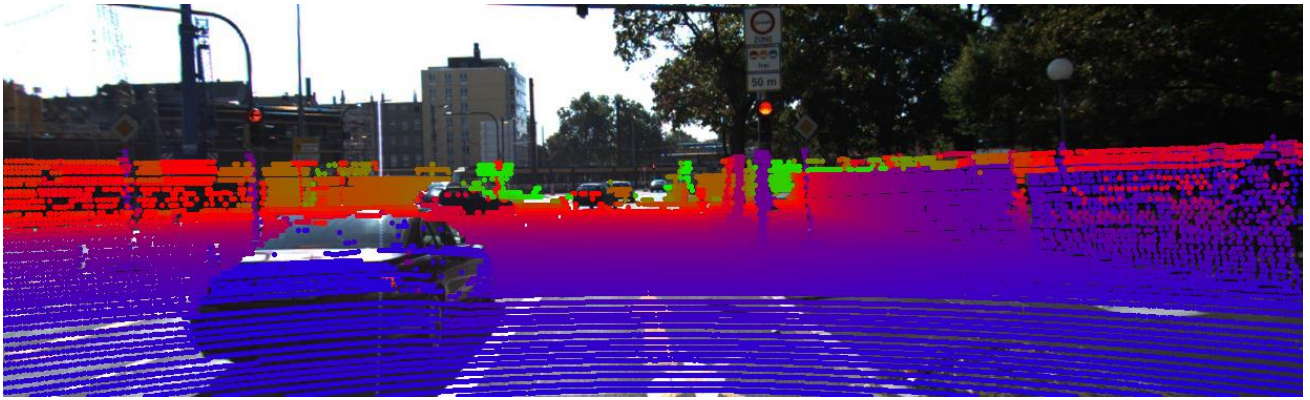
➤ preprocess_3d_point: فرآیند پیش پردازش های مورد نیاز برای نمایش عمق با تصویر RGB را انجام داده و تصویر

حاصل را ذخیره می کند.

خروجی تشخیص عمق برای چهار فریم بصورت زیر قابل مشاهده است. (برای هر نقطه، عمق اندازه گیری شده در قالب رسم

یک دایره کوچک در تصویر که رنگ آن بر حسب اندازه ی آن می باشد، خروجی داده شده است)





ج: درون یابی نقشه عمق

درون یابی یکی از مباحث مهم حوزه آمار و احتمال بوده و روش های متعددی عددی و محاسباتی مختلفی برای آن ارائه شده است. سه روش مهم و اساسی درون یابی که اغلب مورد بررسی قرار می گیرند عبارت است از:

➤ خطی (linear): در این روش درون یابی، ما فرض بر این داریم که تغییرات بین نقطه ها بصورت خطی می باشد و اگر ما مقدار نقطه ای را نمی دانیم و میخواهیم تخمین زنیم، صرفا کافی است که نقاط قبل و بعد آن را در نظر گرفته و با تشکیل یک معادله تغییرات خطی، مقدار آن را بدست آوریم. این روش تقریبا دقت خوب و بسیار مناسبی را ارائه می کند چرا که منطقاً تغییرات خطی در فواصل کم مورد انتظار و تایید است. اما این روش درون یابی یک مشکلی دارد و آن این است که در نواحی حاشیه ای که محیط sparse می باشد نمی توان برای همه نقاط، نقطه قبل و بعد متصور بوده و معادله خط تشکیل داد و این باعث می شود نتوان تخمین برای آن ارائه نمود که ضعف این روش محسوب می شود. پس لذا قابل انتظار و مشاهده خواهد بود که نتیجه تخمین و درون یابی در نواحی مرکزی تصویر با دقت بالا و خوب حاصل شده اما در حاشیه ضعف وجود خواهد داشت.

➤ مکعبی (cubic): این روش مورد فوق می باشد با این تفاوت که تغییرات را صرفا بر اساس نقاط قبل و بعد درون یابی نکرده و توسط مراتب بالاتر این کار را انجام می دهد. به جهت آماری شاید این روش بهتر از خطی باشد اما در کاربرد ما اصلا مناسب نمی باشد زیرا تغییر عمق در در صحنه سه بعدی تابعی از مراتب بالاتر و نقاط با فاصله بیشتر نمی باشد

و پیوستگی بین عمق و تغییرات آن وجود دارد. لذا قابل انتظار و مشاهده خواهد بود که به هیچ عنوان این روش درون یابی مناسب نمی باشد.

➤ نزدیکترین نقطه: در این روش، نزدیک ترین نقطه (توسط معیار مشخصی نظیر اقلیدوسی) که دارای مقدار مشخص می باشد انتخاب شده و مقدار آن برای نقطه‌ی مورد نظر در نظر گرفته می شود (کپی می شود). این روش در نواحی‌ای که نقاط از قبل مشخص زیاد و متراکمی وجود داشته باشد، تقریباً دقت خوبی را ارائه می کند چرا که منطقی است که نقاط نزدیک بهم عمق نزدیک به هم داشته باشند اما این ایده دارای یک مشکل است و آن این است که نقاط نزدیک عمق نزدیک بهم شاید داشته باشند اما اینکه عمق دقیقاً یکسان داشته باشند خارج از توجیه می باشد. لذا می توان انتظار داشت این روش دقت کمتری از روش خطی ارائه می کند اما در مقابل می تواند تمامی نقاط موجود در تصویر (چه حاشیه/چه مرکزی) را درون یابی کند ولو آنکه درون یابی انجام شده از دقت خوبی برخوردار نباشد.

با عنایت به توضیحات فوق و نتایج زیر که برای هر چهار تصویر ورودی، هر سه روش درون یابی انجام شده است، روش خطی به نظر نتیجه دقیق تر با بافت روان تری از عمق را حاصل کرده است اما در نواحی حاشیه ای نتیجه نامطلوبی داشته است. روش نزدیک ترین نقطه نیز با اندکی اختلاف و ضعف در مقابل روش خطی در نواحی مرکزی توانسته درون یابی مناسبی را ارائه کند اما در مقابل برای نواحی حاشیه ای بهتر عمل کرده است. روش مکعبی نیز طبق توضیحات و انتظار قبلی کاملاً تصویر را منهدم ساخته است (: میتوان بدین گونه جمع بندی نمود که روش خطی برای دقت خوب در نواحی مرکزی مناسب بوده و روش نزدیکترین نقطه نیز برای نواحی حاشیه ای نظیر آسمان مناسب بوده است.

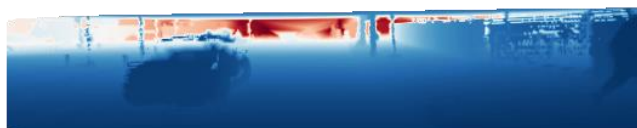
Original depth | img: 0



Interpolation with Nearest method



Interpolation with Linear method



Interpolation with Cubic method



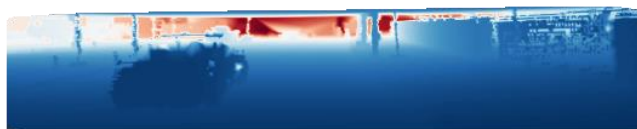
Original depth | img: 1



Interpolation with Nearest method



Interpolation with Linear method



Interpolation with Cubic method



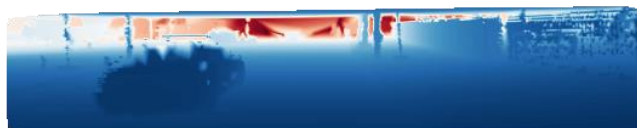
Original depth | img: 2



Interpolation with Nearest method



Interpolation with Linear method



Interpolation with Cubic method



Original depth | img: 3



Interpolation with Nearest method



Interpolation with Linear method



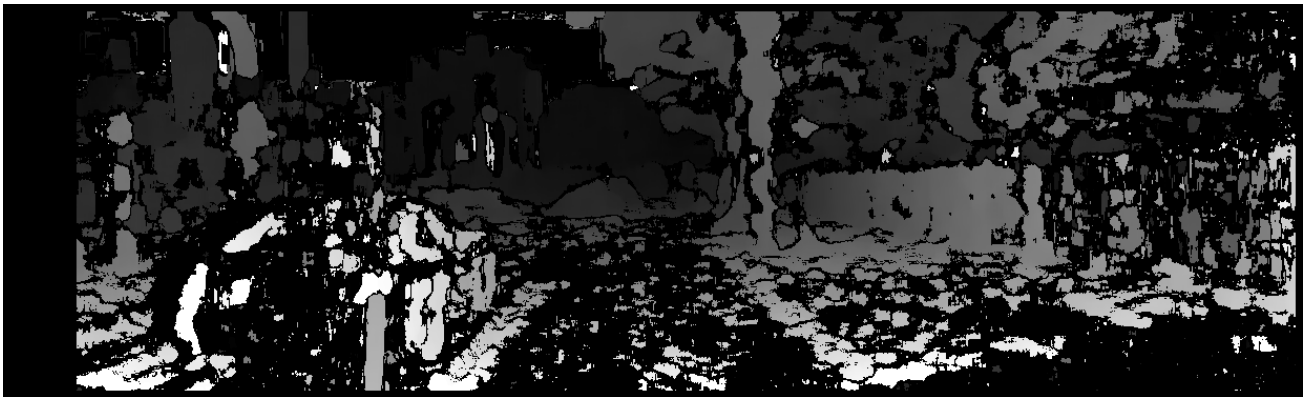
Interpolation with Cubic method



مسئله چهارم

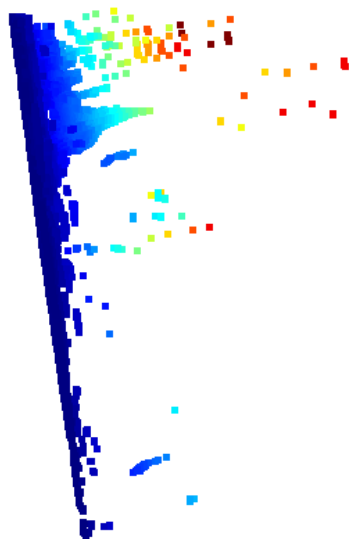
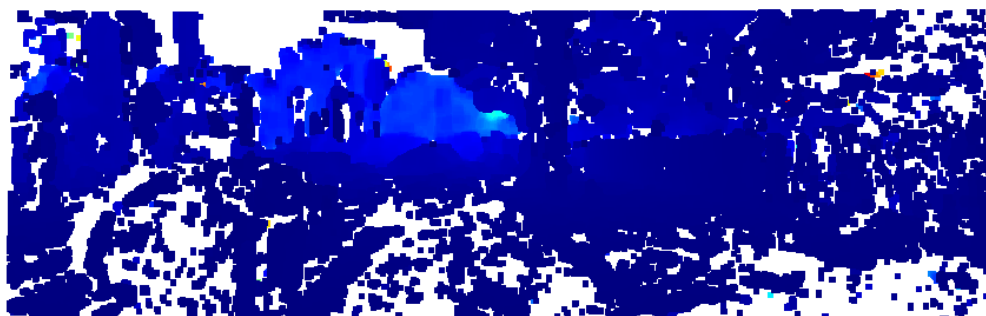
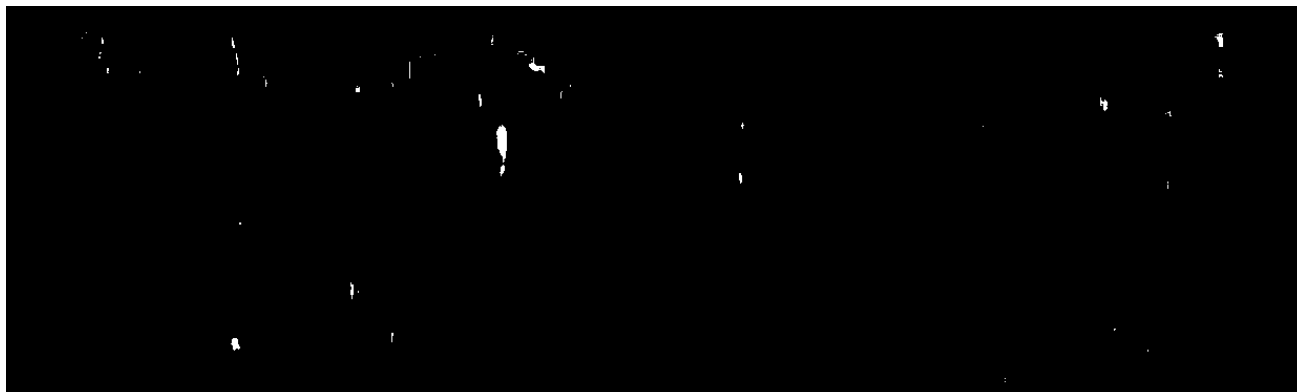
د: تشخیص عمق با روش استریو با دو تصویر و مقایسه آن ابر نقاط

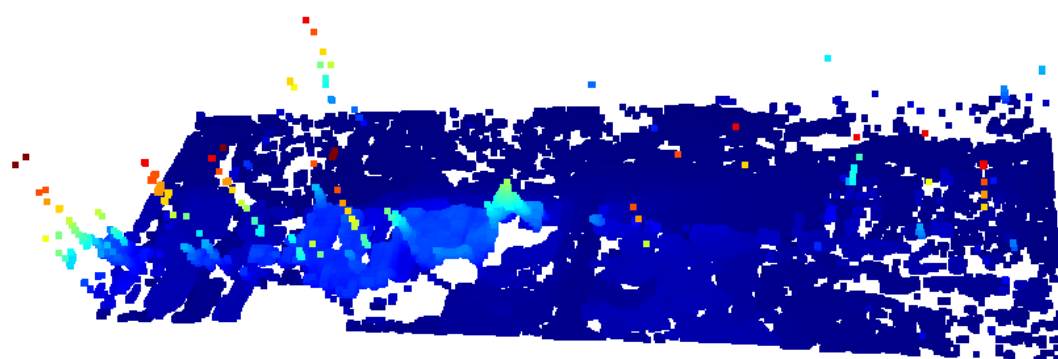
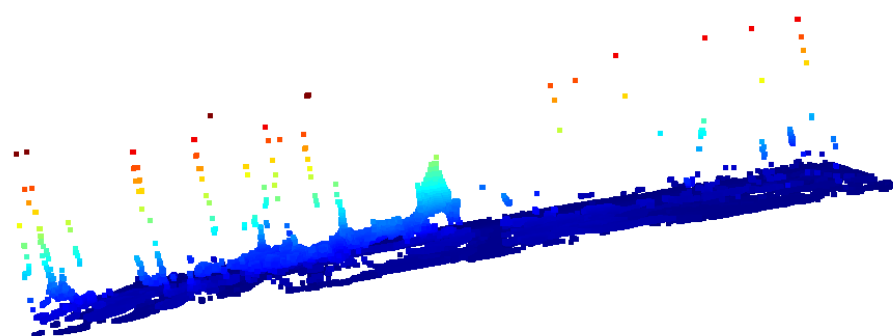
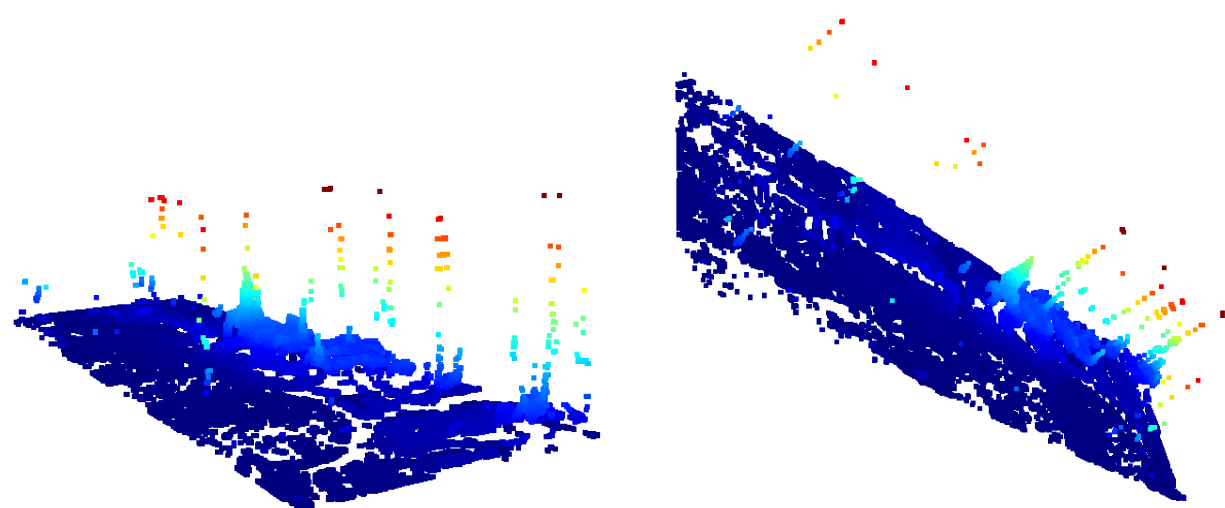
برای تشخیص عمق با روش استریو، از دو تصویر مربوط به یک صحنه استفاده می‌شود که اصطلاحاً به آن دو، تصویر راست و چپ گفته می‌شود. برای ایجاد نقشه disparity از دو تصویر چپ و راست از تابع StereoBM_create از کتابخانه open-cv استفاده شده است که در زیر نتیجه‌ی آن قابل مشاهده می‌باشد.



بر اساس فرمول تدریس شده در کلاس و در نظر گرفتن فاصله کانونی و فاصله خط افقی دوربین‌ها نقشه عمق را بدست آورده و همچنین همانند ابر نقاط در فضای سه بعدی نقاط عمق را نمایش می‌دهیم که بصورت زیر حاصل می‌شود (نقشه عمق نمایش داده شده بصورت دو بعدی چندان قابل تفسیر نمی‌باشد لذا نمایش سه بعدی آن بهتر میتواند اطلاعات مورد نیاز برای مقایسه را فراهم کند)

نمایش دو بعدی و سطح خاکستری نقشه عمق حاصل از استریو:





در مقایسه‌ی نقشه‌ی عمق حاصل از ابر نقاط و استریو قابل مشاهده است که روش استریو بر خلاف ابر نقاط برای همه‌ی نقاط تصویر عمق تخمینی ارائه می‌کند اما این در حالی است که ابرنقاط صرفاً برای بخشی از تصویر که سیگنال ارسالی لایدار بازتاب شده و انعکاس آن دریافت شده است عمق را تخمین زده است ولی قدرت و نکته‌ی مثبت ابر نقاط در دقت بسیار بالاتر در تخمین عمق می‌باشد بطوری که از صحنه سه بعدی درک کاملاً مناسب و نزدیک به واقعیتی را فراهم می‌کند. به صورت خلاصه می‌توان این گونه بیان کرد که برتری ابرنقاط در دقت بسیار بالای تخمین عمق می‌باشد ولو آنکه برای همه نقاط تصویر عمق تخمینی فراهم نکند و در طرف دیگر برتری استریو در فراهم ساختن تخمین عمق برای همه‌ی نقاط تصویر می‌باشد ولو آنکه عمق تخمینی از دقت خوبی برخوردار نباشد.

مسئله پنجم

ه: نمایش سه بعدی صحنه

برای نمایش سه بعدی صحنه ما بایستی برای تمامی پیکسل های تصویر عمق متناظر را داشته باشیم. برای این منظور بنده از درون یابی نقشه عمق با روش خطی استفاده کرده و با ترکیب آن با تصویر دو بعدی رنگی متناظر توسط تابع `create_from_color_and_depth` از کتابخانه `open3d` اقدام به ایجاد تصویر عمق دار می کنیم و در مرحله بعد با ایجاد تصویر `RGBD` و نمایش آن مانند مسئله اول، صحنه را بصورت سه بعدی نمایش می دهیم. در زیر تصاویر مختلفی از نما های متفاوتِ صحنه‌ی سه بعدی بازسازی شده آورده شده است.



