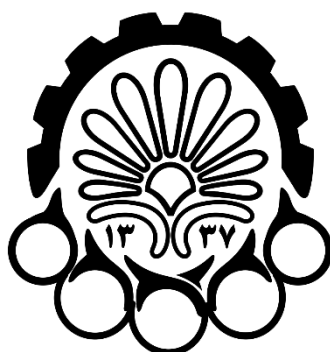


به نام خدا



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

تمرین درس بینایی ماشین - سری ششم

فردین آیار

شماره دانشجویی: ۹۹۱۳۱۰۴۰

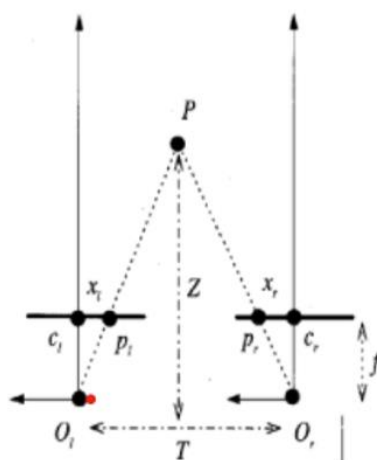
استاد: دکتر صفابخش

(۱) به دلیل مشکل بودن عکس برداری دقیق بدون تجهیزات مناسب، برای این تمرین از دیتاست‌های آماده موجود در اینترنت استفاده می‌کنیم. در شکل ۱، جفت تصویر استریو انتخاب شده نمایش داده شده‌است.



شکل ۱- تصاویر استریو استفاده شده در تمرین

(۲) با توجه به **Error! Reference source not found.** اگر بین دو مثلث  $PO_1O_r$  و  $Pp_1p_r$  از رابطه تشابه مثلث‌ها استفاده کنیم، خواهیم داشت:



$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z} \Rightarrow \begin{cases} Z = f \frac{T}{d} \\ d = x_r - x_l \end{cases}$$

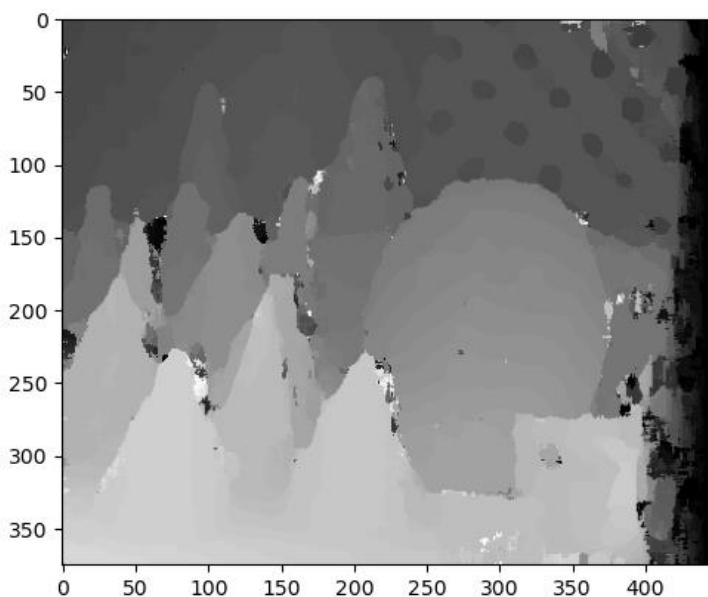
بنابراین برای به دست آوردن عمق هر نقطه در تصویر، کافی است اختلاف محل آن در دو تصویر  $d$  را بدست آوریم. ماتریسی که شامل اختلاف نقاط متناظر در کل تصویر است را اصطلاحاً نقشه عدم توافقی می‌نامند. یک راه رایج برای بدست آوردن نقشه عدم توافقی، استفاده از روش مبتنی بر همبستگی است که در ادامه آن را توضیح می‌دهیم.

در روش مبتنی بر همبستگی، دو پنجره از دو تصویر با هم تطبیق داده می‌شود. پنجره اول در یک نقطه ثابت از تصویر مرجع قرار می‌گیرد و پنجره دیگر در تصویر دیگر جابه‌جا می‌شود. با توجه به ساختار سیستم استریو، کافی است پنجره را تنها به سمت چپ یا راست حرکت دهیم. البته در شرایط عادی بهتر است پنجره را در جهت‌های بالا و پایین نیز حرکت دهیم. میزان جابه‌جایی که یک تابع هزینه بین دو پنجره را حداقل می‌کند، به عنوان عدم توافقی بین دو نقطه در نظر گرفته می‌شود.

۳) کد مربوط به این قسمت در فایل `1.py` قرار دارد. از تابع هزینه مجموع مربعات فاصله به عنوان تابع هزینه استفاده می‌کنیم. با فرض ایده آل بودن سیستم تصویر برداری و به منظور افزایش سرعت الگوریتم، پنجره به ابعاد ۱۵ را تنها در یک راستا و به میزان

شکل ۲- ساختار سیستم استریو

۶۴ پیکسل جابه‌جا می‌کنیم. خروجی در نشان می‌دهد که روش پیاده‌سازی شده عملکرد نسبتاً خوبی دارد.

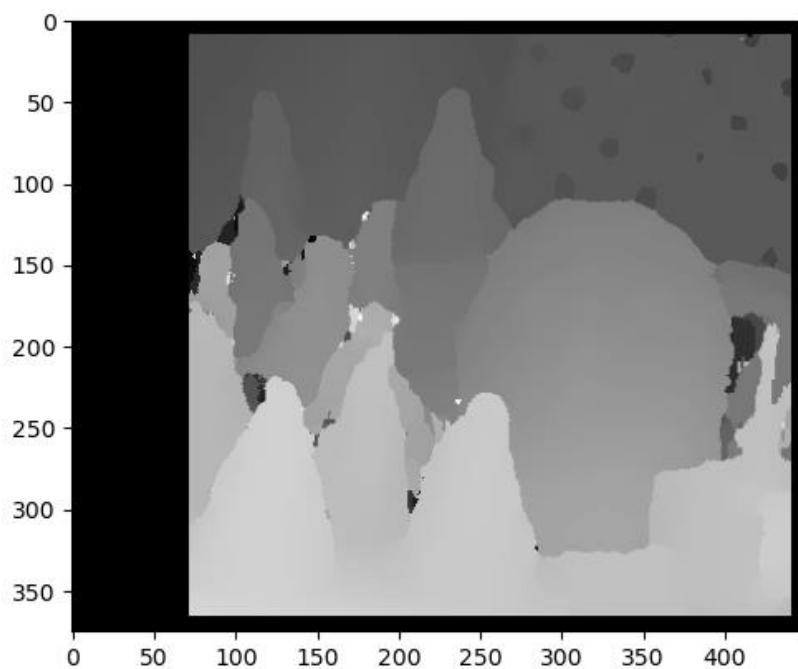


شکل ۳- خروجی الگوریتم محاسبه نقشه عدم توافقی

۴) کد مربوط به این قسمت در فایل `2.py` قرار دارد. ابتدا پارامترهای کلاس `StereoBM` را توضیح می‌دهیم:

- `num_disparities`: حداکثر میزان جابه‌جایی (بازه‌ی جستجو) پنجره برای یافتن نقطه متناظر است. پایین بودن این مقدار باعث می‌شود عدم توافقی نقاطی که جابه‌جایی بیشتری دارند (نقاط نزدیک‌تر)، به درستی محاسبه نشود. بالا بودن این مقدار هزینه محاسباتی الگوریتم را افزایش می‌دهد.
- `texture_threshold`: این پارامتر نقاطی که در همسایگی آن‌ها بافت به میزان کافی برای تطابق وجود ندارد را حذف می‌کند.

- **speckle\_size**: روش مبتنی بر همبستگی ممکن است در لبه‌ها ناحیه‌های کوچک نامتعارف تولید کند. به منظور برطرف کردن این ناحیه‌ها، کلاس StereoBM از یک روش پس پردازشی استفاده می‌کند. پارامتر **speckle\_size** سایز پنجره پس پردازشی را نشان می‌دهد. مقدار مناسب این پارامتر باعث بهبود کیفیت خروجی می‌شود اما هرچه سایز این پنجره بزرگ‌تر باشد، ناحیه‌های بیشتری از تصویر حفظ می‌شود و حتی ممکن است بخش‌های اشتباه از تصویر پاک شود.
  - **uniqueness\_ratio**: این پارامتر مربوط به مرحله پس پردازش خروجی است. در نقشه خروجی اگر پیکسلی به اندازه کافی از پیکسل‌های اطرافش (از نظر تابع هزینه) بهتر نباشد، حذف خواهد شد. به عنوان مثال مقدار ۲ برای این پارامتر یعنی هر پیکسل باید حداقل ۲ درصد از پیکسل‌های اطرافش (از نظر تابع هزینه) بهتر باشد. مقدار بالای این پارامتر، به خصوص در ناحیه‌های هموار، ممکن است خروجی را تخریب کند.
  - **prefilter\_size**: اندازه پنجره در فاز پیش پردازش تصویر را مشخص می‌کند. در این فاز روشنایی و بافت نواحی مختلف اصلاح می‌شود تا نتایج مرحله تطبیق پنجره‌ها بهبود یابد. انتظار می‌رود افزایش اندازه پنجره در عکس‌های بی کیفیت، عملکرد الگوریتم را افزایش دهد. در مورد عکس استفاده شده، مقدار این پارامتر تاثیر محسوسی در خروجی ندارد.
- با توجه به توضیحات فوق، خروجی این روش در شکل ۴ نمایش داده شده‌است. در مقایسه با الگوریتم پیاده‌سازی شده، خروجی کتابخانه OpenCV کیفیت بیشتری دارد و از نظر سرعت اجرا، چندصد برابر سریع‌تر است. لازم به ذکر است تمام پارامترهای توضیح داده شده در این بخش، به جز **num\_disparities** صفر تنظیم شده‌اند.



شکل ۴-خروجی کتابخانه OpenCV