



دانشکده مهندسی کامپیوتر

تمرین سری ۸ بینایی کامپیوتر

نام درس

مبانی بینایی کامپیوتر

نام دانشجو

زهرا انوریان

نام استاد درس

دکتر محمدی

پاییز ۱۳۹۹

سوالات

۱. دو تصویر داریم که تصویر دوم حاصل تبدیل شباهت (similarity) یافته تصویر اول است. فرض کنید نقاط کلیدی متناظر دو تصویر را نیز داریم که ممکن است شامل تناظر اشتباه نیز باشد. مرحله به مرحله توضیح دهید که چگونه می‌توان با کمترین خطا پارامترهای این تبدیل را به دست آوریم (۳۰ نمره)

پاسخ: با استفاده از این تبدیل می‌توان انتقال، چرخش و تغییر مقیاس را بر روی عکس اعمال کرد. تابع تبدیل شباهت در زیر آورده شده است.

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a \cos \theta & -a \sin \theta & t_x \\ a \sin \theta & a \cos \theta & t_y \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

که دارای ۴ پارامتر t_y, t_x برای انتقال و θ برای چرخش و a برای تغییر مقیاس است پس فقط به دو نقطه نیاز داریم ولی ممکن است آن دو نقطه با خطا به یکدیگر متناظر شده باشند و باعث شود که تبدیل دچار خطا شود به همین دلیل از تمام نقاط کلیدی استفاده می‌کنیم تا تبدیل دقیق‌تری داشته باشیم. حال برای اندازه‌گیری خطا می‌توان از حداقل مربعات و بهینه‌سازی استفاده کرد اما این روش به داده‌های پرت حساس است.

حداقل مربعات:

$$Cost = \sum (x_1 * a * \cos \theta - y_1 * a * \sin \theta + t_x - x_2)^2 + (x_1 * a * \sin \theta + y_1 * a * \cos \theta + t_y - y_2)^2$$

بهینه‌سازی: باید نسبت به پارامترها مشتق بگیریم.

$$\frac{d}{dt_x} cost = 2 \sum (x_1 * a * \cos \theta - y_1 * a * \sin \theta + t_x - x_2) = 0$$

$$\frac{d}{dt_y} cost = 2 \sum (x_1 * a * \sin \theta - y_1 * a * \cos \theta + t_y - y_2) = 0$$

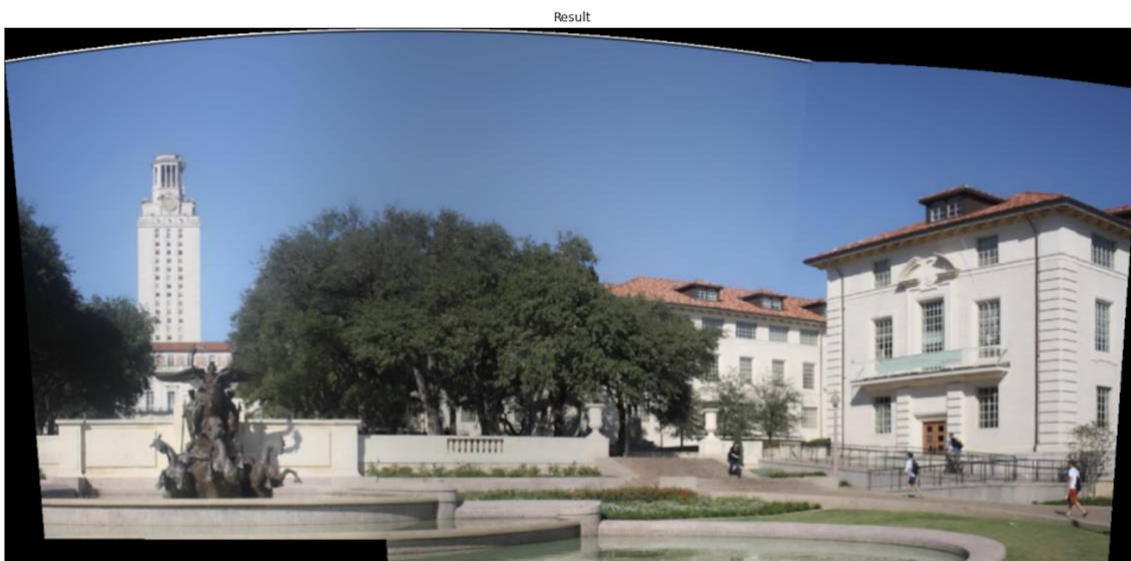
...

حال باید تاثیر داده‌های پرت که به اشتباه نگاشت داده شده‌اند را کم کنیم. برای این کار از الگوریتم RANSAC استفاده می‌کنیم به این صورت که ابتدا دو نقطه از نقاط کلیدی به صورت رندوم انتخاب می‌کنیم و برای آن‌ها تبدیل را محاسبه می‌کنیم و سپس تعداد نقاطی که با این تبدیل به درستی نگاشت پیدا می‌کنند و موافقند را محاسبه می‌کنیم. این مراحل را تا زمانی ادامه می‌دهیم که نقاط موافق به بیشترین تعداد برسند. آنگاه آن تبدیل، تبدیل مورد نظر ما می‌باشد.

۲. یکی از کاربردهای انطباق تصویر ساخت تصاویر پانوراما است. برای این کار از محیط چند عکس گرفته می‌شود و با توجه به روش‌های انطباق تصویر، این عکس‌ها به یکدیگر متصل می‌شوند. برنامه‌ای بنویسید که با استفاده از توابع OpenCV دو عکس 2a.png و 2b.png را به یکدیگر متصل کند (راهنمایی: می‌توانید از تابع cv2.Stitcher_create استفاده کنید. می‌توانید به این [لینک](#) مراجعه کنید). (۳۰ نمره).

پاسخ: با استفاده از تابع cv2.Stitcher_create() یک stitcher با مد PANORAMA می‌سازیم و سپس با فراخوانی تابع stitch آن دو عکس داده شده را می‌دهیم و خروجی عکسی است که از نقاط مشترک آن‌ها به هم متصل شده‌اند و عکس بزرگتری را تشکیل داده‌اند. (خاصیت موزاییک شدن)

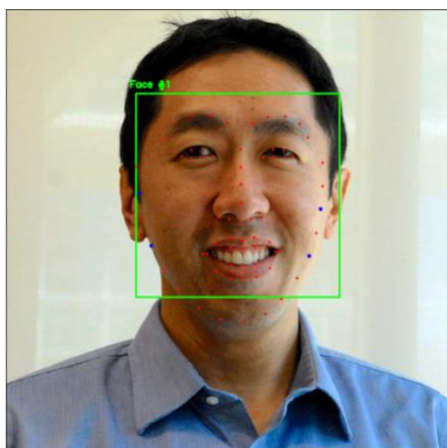
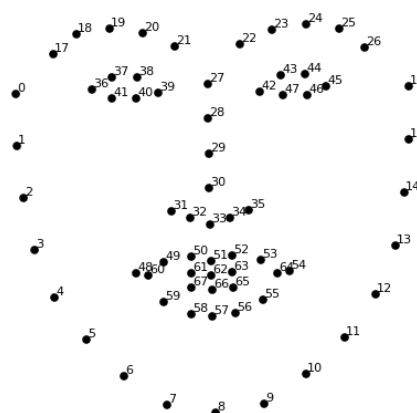
نتیجه:



۳. یکی از راه‌های توصیف چهره استفاده از لندمارک‌ها است؛ که نقاط مشخصی از چهره را مشخص می‌کنند. با کمک لندمارک‌ها و انطباق تصویر، تصویر mask.jpg را به شکل مناسب بر روی چهره‌ی عکس face.jpg قرار دهید (برای کار با لندمارک‌ها می‌توانید از کتابخانه dlib استفاده کنید برای اطلاعات بیشتر به لینک زیر مراجعه فرمایید) (۴۰ نمره).

<http://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python>

پاسخ: برای گذاشتن ماسک بر روی صورت شخص ابتدا باید در تصویر داده شده صورت شخص را تشخیص دهیم و سپس ۴ نقطه‌ای (نقاط آبی) روی صورت شخص با استفاده از لندمارک‌ها مشخص کنیم که ناحیه‌ی مناسبی برای ماسک باشد و با دادن این چهار نقطه به عنوان نقاط عکس مقصد و چهار گوشه‌ی ماسک به عنوان نقاط عکس مبدا به تابع `cv2.getPerspectiveTransform()` می‌توان ماتریس Perspective را بدست آورد. حال باید با استفاده از تابع `cv2.warpPerspective()` تصویر ماسک متناظر بر روی مکانی که بر روی صورت شخص مشخص کردیم را داریم و کفایت ماسک را روی تصویر شخص بیاوریم.



• موفق باشید.