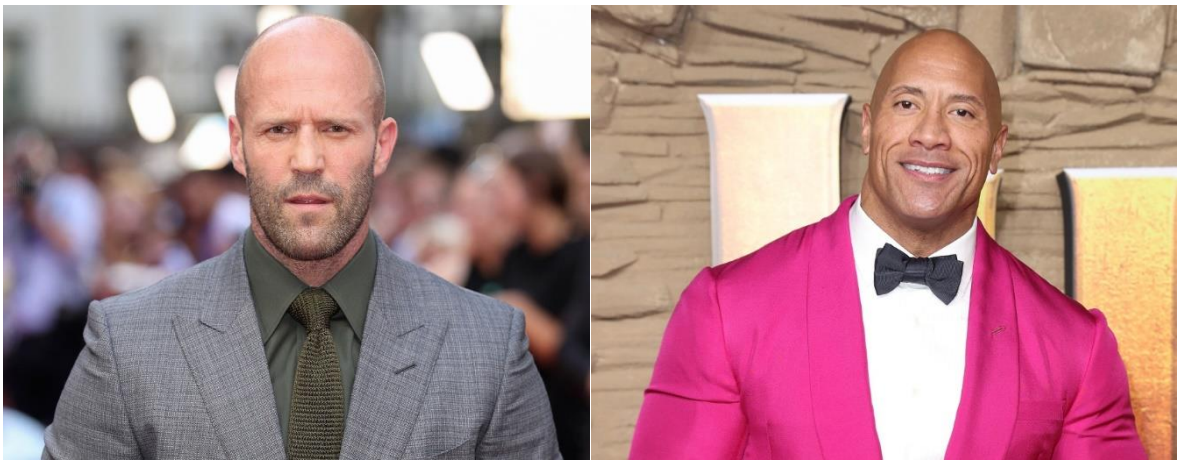
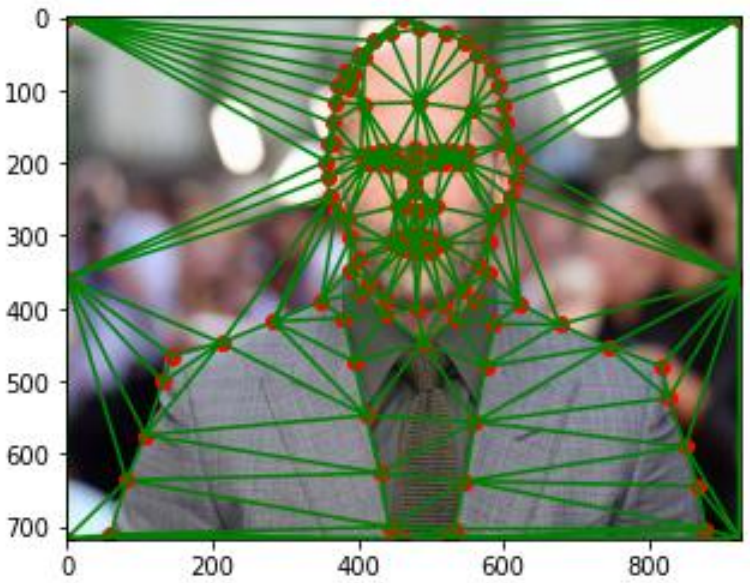
سوال اول

در این سوال قصد داریم تا دو تصویر زیر با با هم morph کنیم.

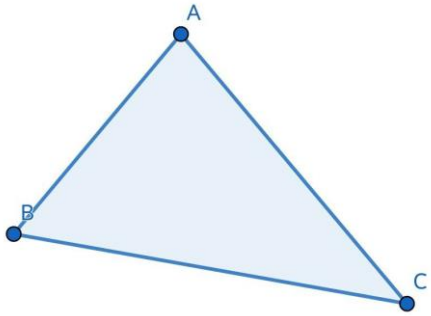


ابتدا تصویر را نقطه گذاری می‌کنیم و بیشتر مکان هایی را انتخاب می‌کنیم که در هر دو تصویر ویژگی مشترک محسوب شوند. مانند چشم، سر، سرشانه، یقه ی پیراهن و ..... در مجموع تعداد نقاط انتخاب شده برای هر عکس 131 است. این نقاط در دو فایل Points1.txt و Points2.txt ذخیره شده اند و حین اجرای برنامه خواهنده می‌شوند. با استفاده از الگوریتم delaunay تصویر سمت راست را بصورت زیر مثلث بندی کرده و از تناظری که بین نقاط موجود در دو عکس وجود دارد استفاده می‌کنیم تا این مثلث بندی را به تصویر سمت چپ هم منتقل کنیم.

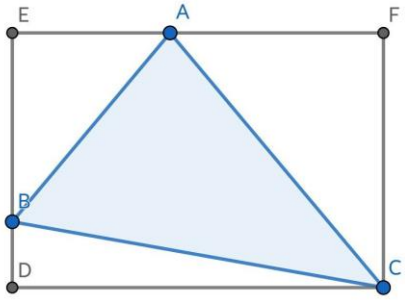


از تابع simplices که در آبجکت حاصل از فراخوانی تابع delaunay ایجاد می‌شود استفاده می‌کنیم تا سه راس هر مثلث را بدست آوریم. با داشتن سه راس مثلث می‌توان تمامی نقاط درون آن مثلث را استخراج کرد. برای این کار از الگوریتم زیر که خودمان پیاده سازی کرده ایم استفاده می‌کنیم.

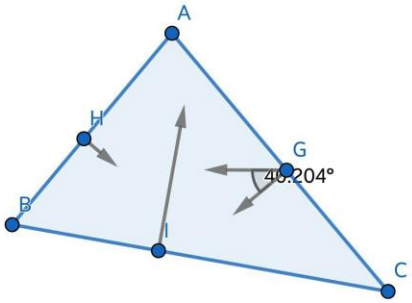
فرض کنید سه نقطه در صفحه بصورت زیر به ما داده شده است:



برای آنکه مجبور نباشیم تمامی نقاط صفحه را چک کنیم، تنها نقاط داخل مستطیلی را که این مثلث را احاطه می‌کند بررسی می‌کنیم:



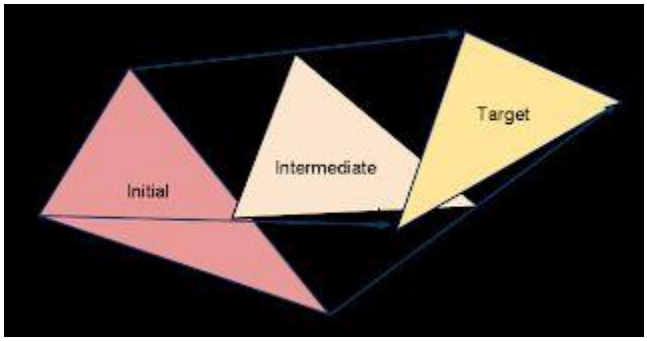
حال باید تعیین کنیم که چه نقاطی داخل مثلث و چه نقاطی خارج مثلث خواهند بود. برای این کار از بردار نرمال ابرصفحه (که در اینجا ابرصفحه ها همان اضلاع مثلث هستند) استفاده می‌کنیم. برای پیدا کردن بردار نرمال کافیست تا جهت هر ضلع را بدانیم و بردار عمود بر آن، بردار نرمال خواهد بود.



همانطور که مشاهده می‌شود در شکل بالا بردار های نرمال هر سه ضلع رسم شده است. حال اگر نقطه ای داخل مثلث باشد، آنگاه اگر روی هر ضلع یک نقطه به دلخواه انتخاب کنیم (ما در پیاده سازی وسط هر ضلع را گرفتیم) و این نقطه ها را به نقطه ی مورد نظر وصل کنیم، آنگاه باید ضرب داخلی این بردار با بردار نرمال ضلع متناظر نامنفی باشد. این اتفاق برای هر سه ضلع باید همزمان

برقرار باشد تا یک نقطه داخل باشد در غیر این صورت خارج مثلث خواهد بود. در تصویر بالا یک نقطه به دلخواه روی هر ضلع نشان داده شده است و بردار های مورد نظر رسم شده است. به این ترتیب توانستیم با یک پیاده سازی ماتریسی خیلی سریع نقاط داخل هر مثلث را پیدا کنیم. این روش در صورتی که جهت مثلث پادساعتگرد باشد راحتتر است و خوشبختانه delaunay این نقاط را پادساعتگرد به ما می‌دهد!

حال فرض کنید بصورت زیر می‌خواهیم یک مثلث میان دو مثلث متناظر در عکس اول و دوم پیدا کنیم. برای پیدا کردن مختصات سه راس این مثلث وسط از ترکیب محدب رئوس مثلث اول و دوم بصورت زیر استفاده می‌کنیم:



$$A = tA_1 + (1-t)A_2$$

$$B = tB_1 + (1-t)B_2$$

$$C = tC_1 + (1-t)C_2$$

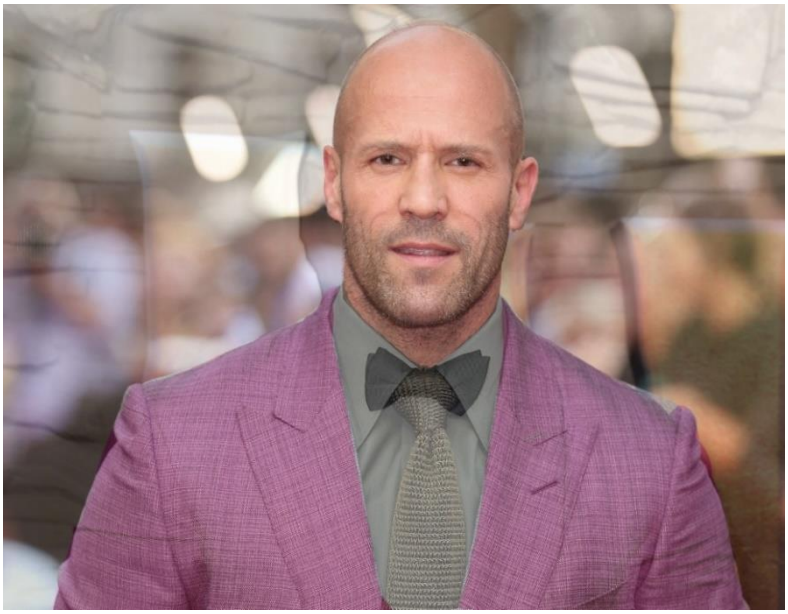
که در اینجا فرض شده است رئوس مثلث وسط ABC، مثلث ابتدای $A_1B_1C_1$ و مثلث انتهایی $A_2B_2C_2$ بوده و $0 \le t \le 1$ است.

حال با استفاده از یک نگاشت affine سعی می‌کنیم این مثلث وسط را به مثلث اول و دوم نگاشت دهیم. جزئیات پیاده سازی این قسمت در سوال سوم تمرین دوم به تفصیل آورده شده است. به این ترتیب می‌توان مثلث اول را به مثلث وسط نگاشت داد. برای این کار از تابعی که برای پیدا کردن نقاط داخل مثلث پیاده کرده بودیم استفاده می‌کنیم و هر نقطه داخل مثلث وسط را با نگاشت معکوس به تصویر اول برده و با یک درونیابی خطی، مقدار پیکسل های آن را پیدا خواهیم کرد. همین کار را برای عکس دوم و عکس وسط نیز انجام خواهیم داد و در انتها این دو تصویر بدست آمده را بصورت ترکیب محدب با یکدیگر cross dissolving خواهیم نمود.

$$I_{ABC} = t\, I^{warped}_{A_1B_1C_1} + (1-t) I^{warped}_{A_2B_2C_2}$$

این الگوریتم را برای تمامی مثلث های موجود انجام می‌دهیم و به این ترتیب یک عکس میان عکس اول و دوم بدست خواهد آمد. این کار را با افزایش t ادامه می‌دهیم تا در انتها به عکس دوم برسیم. طبق صورت سوال این کار در ۴۵ گام انجام خواهد شد و سپس تمامی این عکس ها را به ترتیب عکس قرار داده و از تمامی ۹۰ فریم بدست آمده یک فیلم کوتاه درست می‌کنیم. سرعت پخش فریم ها هم $30\,\frac{frame}{sec}$ قرار می‌دهیم.

در زیر فریم ۱۵ ام و ۳۰ ام به ترتیب آورده شده است:



همانطور که مشاهده می‌شود morphing به خوبی انجام شده است و عکس های میانی به خوبی از ترکیب دو عکس اولیه بدست آمده اند.