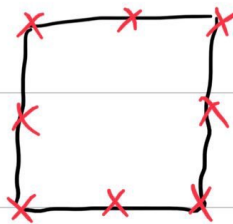


2. اگرچه کلی طایفه این سوال - صورت زیر است:

1) در ابتدا نقاط هم از دو تصویر را مشخص می‌کنیم. به این حالت می‌خواهیم این کار به صورت اتوماتیک صورت گیرد. از تکنیک طلاق استفاده می‌کنیم. به این صورت که به این تکنیک ۶۸ نقطه از نقاط از قبیل و طایفه صورت می‌گیرد. یعنی، هم در... را مشخص می‌کنیم. در ادامه مجموعه‌ی نقاط صورت می‌گیرد. تصویر را هم به این عملیات می‌کنیم. به مجموعه‌ی نقاط لقای می‌کنیم. این نقاط در شکل زیر مشخص شده‌اند:





هم از تانجانه استاده می‌نیم

۳- به ازای هر فریم ورودی  $t$  (فریم  $t$  ام)

- با وزن  $\frac{t}{n}$  و  $\frac{n-t}{n}$  میان فریم‌های قبلی و جاری می‌نیم

- با استفاده از نظایر  $t$  و  $n-t$  میان فریم‌های قبلی و جاری می‌نیم

وزن  $\frac{t}{n}$  (مثلاً اگر  $t=1$  و  $n=2$ ، یعنی میان فریم‌های قبلی و جاری می‌نیم)

وزن  $\frac{n-t}{n}$  (مثلاً اگر  $t=1$  و  $n=2$ ، یعنی میان فریم‌های قبلی و جاری می‌نیم)

مثلاً که از قسمت قبل می‌آید است  $warp$  می‌نیم. (به

این صورت که نقاط متناوب در میان فریم‌های قبلی و جاری می‌نیم)

در بیلان متناوب در میان فریم‌های قبلی و جاری می‌نیم.

- فریم محاسبه شده را به لیست فریم‌ها اضافه می‌کنیم.  
۶- لیست فریم‌ها را به کمک cv.VideoWriter به ترتیب همان‌لیست  
به mp4 تبدیل می‌کنیم. اگر ابعاد True, circular باشد همان‌لیست  
به صورت وارونه توی cv.VideoWriter به ابعاد ویدیو قبلی اضافه می‌شود  
به این ترتیب مورفینگ حلقه‌ای و بی‌پایانی خواهد شد.

نتیجه کار:

رابطه کانفیک به شکل Image Morphing داده می‌شود. به این صورت که  
کانفیک به شکل: اسم فایل مقصدی و اسم فایل مقصدی  
هم ۱- فریم ۱ به ۲- اسم فایل مقصدی هم ۱- است.

در ادامه دو مقودر خواننده می شود و با مقادیر اولیه به ساندن کلاس `ImageMorpher` داده می شود.

ساندن کلاس `ImageMorpher` علاوه بر مقداردهی اولیه متغیرهای کلاس، یک `predictor` و `detector` از کتابخانه `pylib` را هم می سازد. (برای ساختن یک کپی فایل به حجم بار همراه فایل پیوسته به این فایل افعال میل برای یادگیری کتابخانه `pylib` بود).

هم چنین، آبجکت های `BORDE_MODE` و `FLAGS` که برای `Warp` کردن استفاده می شوند مقداردهی می شوند.

در ادامه بخش اصلی که عبارتست از `calculate_morphing` به صورت زیر

امروز می‌شود:

- ۱- رابطه درستی  $calculate\_corresponding\_points$  نقاط متناظر را حساب می‌کند.
- ۲- در  $delannoy\_triangulation$ ، استفاده از متد  $delannoy$  مثلث‌بندی می‌کند محاسبه می‌شود.
- ۳- در این  $delannoy$ ، استفاده از مثلث‌بندی محاسبه می‌کند در  $delannoy\_triangulation$  محاسبه می‌شود.

نکته: ۱:

نقاط محوری هر یک از تقارن محاسبه می‌شود و نقاط متناظر را محاسبه می‌کند. محاسبه می‌شود. (در صورت تغییر اگر نقاط زیاد باشد در صورت فاصله)

بعداً

نقاط حرکتی به بیله می کشند.  $\text{detector}$  محاسب می شود. به کمک آن نقاط را حذف  
می کنند و در نتیجه هم به نقاط حرکتی از نقاط حذف می شود.

صورت ۲:

در این بخش به بیله کش  $\text{SubD}$  از نتایج  $\text{at}$  می کشند و به  
می کشند.

صورت ۳:

در ابتدا به از این هر فریم، ابتدا به استفاده از  $\text{np.zeros}$  می کشیم در ادامه نقاط حرکتی  
را به هم می کشیم و به کمک آن به فریم می کشیم در ادامه به این حرکتی از فریم

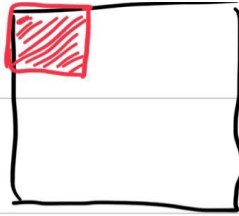
ها با استفاده از `getAffineTransform` و `wrapAffine` متد سر در نظر  
وایس گنجیم (با استفاده از `INTER_LINEAR`)  
نکته: از `BORDER_REFLECT` برای بخش زینت های متد ها استفاده  
شود.

قبل از درج کردن متد ها استفاده از `fillConvexPoly` متد سر  
نظر در مقابل یک متد بدست می آید.  
در ادامه روش متد `build` بنویس به انگیزان `circuler` ویدیو موفقی  
لایه می شود. (با استفاده از `out.VidoeWriter`)



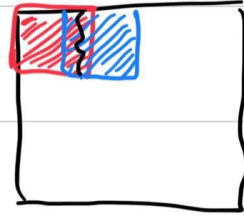
۱ در این مثال تصویر texture که مربوط به دانشگاه است را synthesis می کنیم و از بین روش های مربوط به synthesising (الوارتیم Quilting استفاده می کنیم).  
موردی که این روش را موجب می شود همبستگی بین این ها را عبور در تصویر texture است. به خلاف روش احتمالی این روش تصویر synthesis شده را یک بار به یک بار می سازد. (نه یکبار یکبار)

مراحل ساده تر تصویرهای Synthesis شده به صورت زیر است:  
در ابتدا یک برگ به صورت اتفاقی و مختلفی از یک قسمت از تصویر منبع را در قسمت و به بالا می کشیم. به این ترتیب تصویر Synthesis شده به صورت زیر خواهد بود:



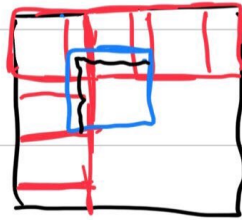
در ادامه ردیف اول تقویم Synthesis را کامل می کنیم به این ترتیب  
که این بین کننده های مورد برای بلوک سمت راست بلوک قبلی رنگ  
(رنگین لیمو زردین ها به آن) یک مورد را به صورت تقاضای انتخاب می کنیم  
به ازای یک مقدار طول اولی که در استیبل می باشد Image Synthesis داده شده  
است. حال در سمت اولی دو بلوک انجام گرفته و به عنوان ۲ می رانیم تا این  
صفت و منفی ضعیف شود در نهایت باید بلوک کمینه که باقی می ماند اولی  
را به این کن وصل می کنند و این را می رانیم. پس این فین آن (به صورت تقریبی) است

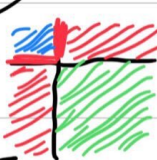
صِدِّ قَسَمَتِ اَوَّلِیِّ کہ با بیٹیں برابر ہے اسے رازِ بدستِ اہلِ وسعت  
 اسے آں رازِ بدستِ سمیت اسے آں رنگِ انتخابِ مکی تحیم۔ ہر لہجہ  
 صورتِ تقدیرِ Synthesis بہ صورتِ زیرِ فواہدہ بود:



بالادہ سے لہجہ روزِ مدنیہ اولیٰ کمالی سی شد فتحِ سنین بہ صورتِ مثالیہ اولیٰ  
 بکبرِ طرہامِ ہم کمالی سی شد در ادبہ برای بکبرِ ہاں کہ بہ ترتیب  
 ۸۰۰ لکھ ہفت ہفت لکھ لکھ در دو لکھت با ۵۰ صِدِّ بکبر

هزینه کاهش انتخاب کنیم. داده‌های این روند داریم.



که در این فرآیند در سمت چپ، به یاد قسمت‌های زمانی |  می‌پردازیم. این کار با استفاده از ۳۲ بایت کد انتخاب می‌شود. در نهایت، فرآیند Synthesis پس از کدابی می‌شود و بایت‌ها در سطرها و ستون‌ها  
نمایش داده می‌شود.

در این مقایسه، ما به ساندویچ‌ها Image Synthesizer داده می‌شود تا با مقادیر

اولی داده شده است: synthesizer ساخته می شود.

داده شده `synthesis_input_image` و `synthesis_image` به عنوان ورودی می باشد.

محاسبه می کند. خروجی شده که به صورت زیر است:

- ۱- در ابتدا `synthesis_image` را به `synthesis_input_image` می دهد.
- ۲- داده به ازای `row, col` به ترتیب به `synthesis_image` می دهد.

`row, col` محاسبه می شود در مکان مناسب `synthesis_image` می دهد.

در `vertical_opt_cut` به ترتیب `row, col` به `synthesis_image` می دهد.

به ترتیب `row, col` به `synthesis_image` می دهد.

به ترتیب `row, col` به `synthesis_image` می دهد.

به ترتیب `row, col` به `synthesis_image` می دهد.

یہ دراصل اس بات کو ظاہر کرتا ہے کہ اگر  $\text{Opt\_Block\_Error} < \text{threshold}$  ہو تو اس بلاک کو منتخب کرنا چاہیے۔

دوسرے درجے میں اس بلاک کو منتخب کرنا چاہیے جس کا  $\text{Opt\_Block\_Error}$  کم ہو۔

$(1 + \text{threshold}) \times \text{Opt\_Block\_Error}$

اس بات کو مدنظر رکھ کر اس بلاک کو منتخب کرنا چاہیے جس کا  $(1 + \text{threshold}) \times \text{Opt\_Block\_Error}$  کم ہو۔