

Assignment 2

Qi Le 5674954

Sift Feature Extraction and SIFT Feature Matching

def find_match(template, target): The two grey scale images are passed to this function and SIFT module of OpenCV is used to extract keypoints and keypoint descriptors, which are 128 dimension vectors. Then kNN is applied to these points from two different images to get closest neighbors' pairs. The threshold is set to 0.7



Feature-based Image Alignment

def align_image_using_feature(x1, x2, ransac_thr, ransac_iter): The closest neighbors' pairs are passed to this function and the best Affine transform matrix will be returned. First, three correspondences (6 points) will be chosen, and the corresponding affine transform matrix will be calculated based on these three points. Then we back substitute the affine transform matrix to all the template points and apply the RANSAC algorithm to find the best Affine transform matrix fitted for most points. Ransac_thr and ransac_iter are set to 9 and 500, respectively.

Image Warping

def warp_image(img, A, output_size): We use the affine transform to warp the target image to the template using the inverse mapping with bilinear interpolation (interp function imported from scipy.interpolate). This intern function is time consuming.

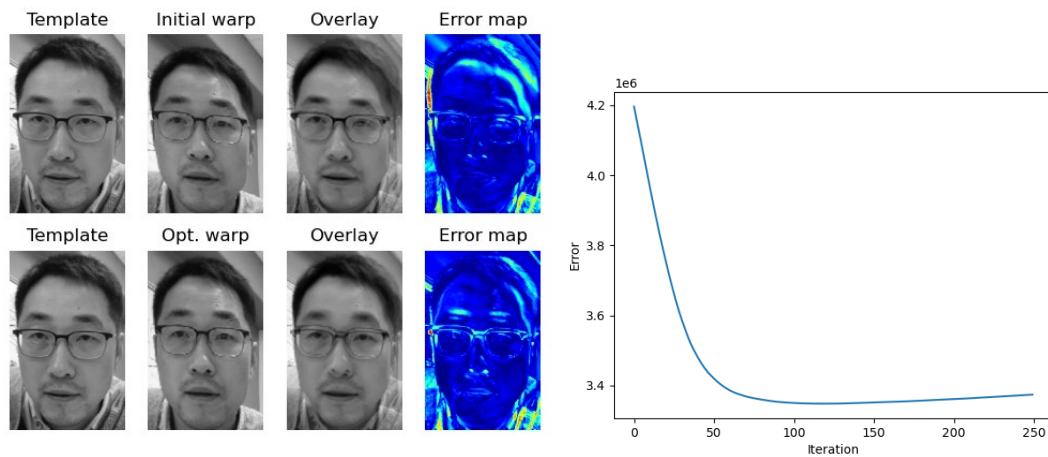


Inverse Compositional Image Alignment

def align_image(template, target, A): We use the initial estimate of the affine transform to align next image using inverse compositional method. This function will return the refined Affine transform matrix in the end. In this function, we first generate the gradient of the image. Then we use the gradient of the image and the Jacobian matrix

to form the steepest decent images, which leads us to the 6×6 Hessian matrix. Then the error between the template image and the warp image will be calculated and the norm of error Δp is generated from inverse Hessian matrix and the sum of the steepest decent images times the error. The improved Affine transform matrix can be calculated from the affine transform of the Δp and itself.

This process will terminate when the error is below the threshold (2000) or the iteration times over 250. In the end, the improved Affine transform matrix will be returned.



Multiframe Tracking

def track_multi_frames(template, img_list): From a template and a set of consecutive images, we initialize the affine transform using the feature based alignment and then track over frames using the inverse compositional image alignment. We repeat the above procedures for all four images.

