CP467 Course Project
Image Processing and Recognition

Duc Minh Nguyen - 203009140

Quang Quynh Anh Lam - 190712100

Mohiuddin Usama - 212090340

Introduction

In this CP467 project, we will implement image processing applications such as feature detection, object identification and image stitching on a dataset of 15 individual objects, and 20 different shots of a cluttered scene. We aim to implement these techniques in practical scenarios using the materials we have learned from CP467, specifically to create a cluttered scene, detect all objects in it using Scale-Invariant Feature Transform (SIFT) on the frontal shot of each object; and partial images into one large scene using Oriented FAST and Rotated BRIEF (ORB).

Regarding details of our scenes, there are 4 full scenes that include all objects, one clear frontal, one zoomed far out, one slightly slanted northward, and last one moderately slanted southwestward, respectively named as S1 to S4. The remaining scenes are cropped and overlapped images of these 4 full scenes.

Please refer to Table 1 below for details of our objects.

Table 1. Annotations of All Indexed Objects in Dataset versus Their Corresponding Names

Indexed Object	Corresponding Object Name
O1	checkered phone case
O2	purple keyboard
O3	almond blossom
O4	LAURIER brochure
O5	BOTTLE JOY strip
O6	CASIO calculator
O7A	skull image
O8	white tiger image
O9	VICHY packet
O10	sledding fox sticker
O11	Fugacar
O12	KEYS packet
O13	GRR red card
O14	GRR yellow card

O15	ORIGINS green tube

Literature Review

For the past method(s) to perform the 2 tasks + Advantages and Disadvantages, we will first talk about task 1 and how the Canny/Sobel Edge detectors which are probably one of the oldest ways of detecting object in images using their edges. They are used for Identifying sharp contrasts in images to find the boundaries of objects. Their advantages are that they are Effective at detecting just the edges in the image and can detect images with noise as well. However, their disadvantages are that they are Inefficient for detecting actual objects in the image because edge detection is not sufficient for object detection and that they Don't provide information about the edge features that could be used for matching.

Moving onto Corner Detection (Harris Corner Detectors) that can be used in task 1. They are used to detect corners since corners are very valuable in image detection. Their advantages are that they are Efficient at detecting corners in the images for specific objects. While their disadvantages are that they are very limited and cannot define objects with just corners, especially for objects that don't have corners and are rounder.

Now moving onto Feature Descriptor Methods (SIFT, ORB), these are used to Extract key points from images and compute their descriptors for matching. Their advantages are that they are robust to changes when it comes to scaling and rotation of the images. Their disadvantages are that They can we computationally intensive, and have a long run time, and due to this they are very inefficient for large datasets. They also struggle with images that have high levels of noise.

Now when it comes to Task 2 for past methods, we have Direct Pixel Matching. This is used to Align Images based on comparing their intensities. Their advantages are that they are simple and efficient. Meanwhile their disadvantages are that they Has a poor performance for different image scaling and different exposures (such as objects overlapping).

We also have feature descriptor-based method such as Sift, SURF, and match algorithms to align images based on detected features (Using Stitcher). Their advantages are they are robust to multiple image transformations and their disadvantages are that the run time is very slow if you are stitching a large dataset of images. The images need to be very similar and have more than a 30% overlap otherwise it's hard to detect the images.

Now we can more into details about the current methods to perform the 2 tasks talking about their advantages and disadvantages. First of all, for task 1 we have Deep learning Methods, there are multiple deep learning methods such as CNNS (Convolutional Neural Networks), YOLO (You look only once) and SSD's (single shot Multibook detector), GANS (Generative Adversarial Networks) and End to End learning. Their advantages are that they are highly accurate and capable of detecting objects in real time, and Robust to scaling, different angles (rotations), different lighting. However, their disadvantages are that they require a large dataset of images since it's a machine learning and deep learning model. They can also be computationally expensive depending on the dataset and it can be hard to debug issues since you need to know what the model is learning.

Now when it comes to task 2, we have seamless blending methods which include algorithms such as graph cutting algorithms that help find the optimal seam to stitch images by reducing the differences along the seam. Their advantages are that can easily stitch the images and automate the entire stitching pipeline meanwhile their disadvantages are that they require large amounts of training data and computational power.

We also have Deep learning methods which we can use for task 2 such as End to End Learning (using neural networks to stitch the images together), and their advantages are that they are Capable of learning patterns in the images to identify them, they can also be used to automate the stitching process. However, when it comes to their disadvantage their main ones are related to having a large dataset to get a better and accurate result otherwise it will have a hard time determining the object. With a large dataset comes high computational power and longer times, so another disadvantage will be the cost of increasing the computational power to reduce the time.

Now when it comes to the group decision on choosing for which methods to use for task 1 and task 2, we chose the following. When it came to task 1, we ended up choosing SIFT to find the interest points and descriptors. Then we chose BF Matching to help match the object images to the scene image to help identify the object in the image. And then we used homography to find the location of the object in the scene image. And when it mainly came to task 2 that required us to stitch all the scene images together, we ended up using algorithms such as Orb Sift and Cropping. We tried Sift and Orb Simultaneously, but Orb was giving slightly better results for all 20 scene images and was able to stitch the images in less run time. However, when it ended up stitching the images, there would be a lot of black patches surrounding the stitched image and to reduce that, we ended up using cropping by converting the stitched image to grayscale, and thresholding the binary value of 0 so that it would remove the black patches in the image. To finalize this is how we were able to do both task 1 and task 2 compared to all the past and current methods provided above in the terminology.

Algorithms

Task 1: requires the applications of feature detection and matching. For feature detection, we implement a combination of SIFT, Harris Corner and Orb.

For matching, we first go through SIFT process alongside with changing matcher thresholds manually, until we reach desired results.

- SIFT: this algorithm detects and describes local features in image by identifying key points, distinctive points resilient to change in scale, rotation and lighting
- Brute Force Matcher: this compares descriptors from two sets of features by calculating distances between them, typically use best matches between key points in different images

Task 2: requires image stitching application

- First, we use Sticher, however this algorithm would output a very large size images, therefore this approach would not be viable and not enough memory could be allocated
- Then, we find the keypoints and descriptors by using SIFT and then matched them using FLANN BASED MATCHER

Through implementation, we have experienced many challenges with our chosen methods, as follows:

1. **Object Selection:** Fundamental understanding on the course materials does not really help us derive the perfect algorithm. We had to go through many different datasets of scenes to filter out the objects that often yield high accurate detection rate with SIFT specifically. We concluded that flat, pronounced-edged (ideally rectangular) and distinctly colorful objects are the most detectable; while spherical or sculpted objects which provide drastically different images in different angles, or single-colored objects which sometimes make insignificant points of interest, they are the hardest ones to detect.
2. **Camera Angle Selection:** Image of objects in a particular scene are heavily impacted by camera angles. Different shots can result in different object shapes and colors, which might eventually make the matching values very much different from the given dataset. Camera angles are also hard to keep track of (ie. for repeating good angles or for successfully avoiding bad angles).
3. **Lighting:** Similar to camera angles, lighting is easily inconsistent in each scene taken, due to various factors such as the change of outdoor lighting (if involved), shadows of adjacent objects, negligent shadows of photographer, and noise values from both object backgrounds and scene backgrounds.

Results

Table 2. Table of Results for Task 1

Abbreviations:

TP = True Positive; FP = False Positive; TN = True Negative; FN = False Negative;

P = Precision; R = Recall; F1 = F1-score; A = Accuracy.

Scene	TP	FP	TN	FN	P	R	F1	A
Scene_1	14	0	1	0	1.00	0.93	0.97	0.93
Scene_2	14	0	1	0	1.00	0.93	0.97	0.93
Scene_3	12	0	3	0	1.00	0.80	0.89	0.80
Scene_4	12	0	3	0	1.00	0.80	0.89	0.80
Scene_5	14	0	1	0	1.00	0.93	0.97	0.93
Scene_6	6	1	1	0	0.86	0.86	0.86	0.75
Scene_7	3	3	2	0	0.50	0.60	0.55	0.38
Scene_8	9	0	1	0	1.00	0.90	0.95	0.90
Scene_9	12	0	2	0	1.00	0.86	0.92	0.86
Scene_10	4	2	0	0	0.67	1.00	0.80	0.67
Scene_11	10	0	2	0	1.00	0.83	0.91	0.83
Scene_12	7	2	0	0	0.78	1.00	0.88	0.78
Scene_13	8	1	1	0	0.89	0.89	0.89	0.80
Scene_14	4	1	0	0	0.80	1.00	0.89	0.80
Scene_15	3	1	2	0	0.75	0.60	0.67	0.50
Scene_16	10	0	3	0	1.00	0.77	0.87	0.77
Scene_17	4	1	0	0	0.80	1.00	0.89	0.80
Scene_18	4	4	0	0	0.50	1.00	0.67	0.50
Scene_19	11	0	3	0	1.00	0.79	0.88	0.79
Scene_20	10	0	3	0	1.00	0.77	0.87	0.77

- Detect objects in task 1 using SIFT
- Stitch objects in task 2 using ORB
- Remove the background of individual objects using www.remove.bg