CP467 - Image Processing & Recognition
Final Project
December 2, 2023
Alexander Torzsas - 210233050 | Shane Shamku - 210315940 | Javier Rodrigues - 210631740
Dr. Zia Ud Din

Literature Review

The evolution of object detection and image stitching in computer vision reflects a journey of remarkable innovation and technological advancement. Let's delve into a literature review of these fields, highlighting key methods and their respective advantages and disadvantages.

Object Detection

Object detection is the practice of attempting to identify objects spatially within an image or video and is often used in medical imaging, facial recognition and video tracking software.

Early Methods: The inception of object detection dates back to the 1960s, with applications in character pattern recognition and later in drug classification and ATM bill recognition.(1) These early methods, including template matching and sliding window approaches, were fundamental yet had limitations, especially in handling variations in pose, size, scale, or occlusions.(1)

2000s: A significant advancement came in 2001 with the Viola-Jones detector, designed for face detection. It introduced a method involving feature extraction and classification, including stages like selecting Haar-like features, creating an integral image, AdaBoost training, and cascading classifiers.(1) While highly accurate, it was constrained in its applicability to other forms of object detection. Furthermore, gaining popularity in 2005, HOG was pivotal for human detection in applications like self-driving cars. It involved extracting contrasts in image regions based on gradient vectors, improving accuracy through local normalization and contrast normalization.(1) However, HOG was best suited for human detection and less versatile for other objects.

Modern Deep Learning Approaches: Nowadays, the advent of deep learning has taken over object detection. AlexNet, emerging from the ImageNet challenge in 2012, marked a turning point with its CNN-based model. Following it, R-CNN and its variants (Fast R-CNN, Faster R-CNN) offered more efficient region proposal and classification mechanisms. (1) These models relied on training neural networks to provide accurate detection. Furthermore, YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector) further enhanced real-time detection capabilities, though they faced challenges in detecting small or clustered objects. (1)

Image Stitching

Image stitching is the process of joining smaller/partial images into a larger composite image and is often used to create panoramas and digital maps. Image stitching is generally composed of three distinct processes, calibration, registration and blending and historically there have been two primary methodologies for performing stitching, direct methods and feature-based methods. (2)

Image stitching began with direct methods, focusing on pixel intensity alignment. Techniques like Cross-Correlation, Sum of Absolute Differences (SAD), and Sum of Squared Differences (SSD) were foundational. (2) These methods, while simple, were susceptible to variations in lighting and required a good initial alignment.

More modern and robust, feature-based methods detect and match distinctive image features such as edges and corners. They are less sensitive to lighting changes and can handle larger displacements. These methods essentially studied the objects they were supposed to find and matched them to their scenes by looking for common distinct features. Some common feature based registration methods include SIFT (Scale Invariant Feature Transform) , SURF (Speeded Up Robust Features) and ORB (Oriented FAST and Rotated BRIEF). (2) The development of advanced feature descriptors significantly improved the efficiency and accuracy of image stitching. However, feature-based methods require a high density of distinct features on objects and are also still susceptible to lighting discrepancies to a degree. High amounts of contrast, glare or other lighting issues can throw off a feature-based detection algorithm.

Finally, advancements in computational power and algorithmic efficiency have led to real-time stitching capabilities, essential for applications like panoramic photography and virtual reality. (2) The integration of machine learning and AI further enhances the adaptability and precision of stitching algorithms. However, these models often have a high computational load and require significant specific training to be successful.

Methods & Algorithm Discussion:

Object detection - This Python code, utilizing OpenCV, creates an object detection system that begins by loading template images of various objects to match with objects in scene images. The detection process includes image preprocessing (grayscale conversion, resizing, Gaussian blurring, thresholding, and morphological operations) to enhance object features. It then employs contour detection and bounding box drawing, with non-maximum suppression to refine these boxes. Object recognition is achieved using the Scale-Invariant Feature Transform (SIFT) algorithm and FLANN-based matcher, where good matches are identified through a distance ratio test and homography transformations are used to pinpoint object locations in scenes. The system annotates detected objects and calculates performance metrics like precision, recall, F1-score, and accuracy. This approach effectively detects and recognizes multiple objects in complex scenes, showcasing the synergy of contour-based and feature-based detection methods.

Keypoints - Our code employs the OpenCV library to detect keypoints in images of objects and scenes. It uses Contrast Limited Adaptive Histogram Equalization (CLAHE) for image enhancement, improving feature visibility. The core algorithm for keypoint detection is the Scale-Invariant Feature Transform (SIFT), known for its effectiveness in identifying and describing local features in images regardless of scale or orientation. The script iteratively processes images from two categories—objects and scenes—extracting keypoints, visualizing them on the original images, and saving the output. This method is particularly useful for tasks like object recognition and image stitching in computer vision.

Matches - This Python code, utilizing OpenCV, is designed for matching keypoints between object and scene images. It first loads specific scene and object images, ensuring they are loaded correctly. The script then employs the Scale-Invariant Feature Transform (SIFT) algorithm to detect and compute keypoints and their descriptors in both images. The key feature matching is performed using the FLANN (Fast Library for Approximate Nearest Neighbors) based matcher, a robust and efficient method for finding correspondences between descriptors. To refine the matches, Lowe's ratio test is applied, filtering out weaker matches and retaining only the stronger, more reliable ones. The resultant good matches are visualized by drawing them on the combined scene and object images using drawMatchesKnn. The script dynamically names and saves these matched images in a designated folder, providing a clear visual representation of the matching process between specific objects and their presence in scenes. This method is particularly useful in applications like object tracking and recognition in complex scenes.

Stitching - This Python code uses OpenCV for image stitching to create panoramas. It starts by loading and converting images to grayscale for feature detection with the Scale-Invariant Feature Transform (SIFT) algorithm. Key features between images are matched using the Brute-Force Matcher, and a ratio test filters out weak matches. The core of stitching is the calculation of homography, which aligns and warps images into a common coordinate system. Each image is sequentially warped and stitched, creating a single panoramic image. This technique is effective for generating wide-angle panoramic views in applications like landscape photography and virtual tours.

Table of Results

Scene	True Positives	False Positives	False Negatives	True Negatives
S1	15	5	1	3
S2	10	5	3	3
S3	12	3	2	4
S4	12	4	1	4
S5	7	7	2	5
S6	5	8	2	6
S7	9	5	1	6
S8	10	3	3	5
S9	9	6	3	3
S10	10	6	3	2
S11	17	3	2	1
S12	13	2	0	6
S13	18	5	2	3
S14	13	2	3	3
S15	12	3	1	5
S16	11	2	1	7
S17	11	4	2	4
S18	11	4	3	3
S19	10	5	1	5
S20	15	1	1	4
S21	17	0	2	2
S22	18	3	0	0
S23	12	5	1	3
S24	11	4	3	3
S25	13	1	1	5

S26	11	2	2	6
S27	17	1	1	3
S28	16	2	1	3
S29	12	0	1	5

Overall Results

Precision: 0.78
Recall: 0.88
F1 Score: 0.83
Accuracy: 0.77

Challenges Faced and Addressed:

In the development of our feature matching program, we encountered several challenges. Notably, the precision of feature matching was critical; we used the SIFT algorithm due to its robustness against scale and rotation variations, yet the FLANN-based matcher initially returned a mixture of correct and false matches. By implementing Lowe's ratio test, we filtered out less reliable matches, significantly enhancing the accuracy of our object recognition. Additionally, the dynamic naming of output files based on extracted metadata required a reliable method to parse image filenames. We addressed this by employing a regular expression search to accurately discern scene and object identifiers, which proved to be effective. Lastly, we programmatically ensured the creation of a dedicated output directory, mitigating any file system errors related to nonexistent paths. These methodical approaches to the challenges have streamlined our workflow, resulting in a robust matching system that reliably identifies and visualizes object correspondences between images.

In our panorama stitching task, we faced several challenges, particularly with variations in lighting, capture distances, and image overlaps. Varying lighting conditions were normalized through histogram equalization and multi-band blending, which smoothed the transitions between images. Discrepancies in capture distances were addressed by scaling images based on feature point correspondences, allowing for consistent sizing across the scene. Finally, inconsistent overlaps were managed by refining feature detection and optimizing the seam-finding algorithm, ensuring that each image segment merged seamlessly. Though these addressed issues seem to be successful, we were still unable to successfully stitch the images to create a larger panorama.

References

V Megha, K K Rajkumar. (2022). A Comparative Study on Different Image Stitching Techniques. Retrieved from, https://ijettjournal.org/archive/ijett-v70i4p205

Borah, C. (2020). A Comparative Study on Different Image Stitching Techniques. Retrieved from, https://medium.com/analytics-vidhya/evolution-of-object-detection-582259d2aa9b