Assignment 2: Homography Estimation, Panorama Stitching, and Augmented Reality

Student Name: Arda Ceylan

Student Number: 2220356041

Panorama Result, AR Video Result and Demo Presentation:
https://drive.google.com/drive/folders/14jEV8tuVlrq5qT53wHdX0VQrAtHFHaNd?usp=sharing

1. Overview

The goal of this assignment was to develop a deep understanding of the geometric relationship between images by implementing a complete pipeline for homography estimation. This pipeline involves detecting and matching robust features (SIFT, ORB), calculating the 3x3 homography matrix using a manually implemented Direct Linear Transform (DLT) algorithm, and achieving robustness to outliers through a RANdom SAmple Consensus (RANSAC) framework.

This core pipeline was applied to two main tasks: first, stitching multiple images of a planar scene into a panorama by warping them into a common coordinate frame and blending the overlaps; and second, building a simple Augmented Reality (AR) application. This AR app projects a source video onto a moving planar surface (a book cover) in a target video, demonstrating the real-world utility of per-frame homography estimation for dynamic geometric alignment.

2. Dataset & Setup

The project uses two datasets provided in the assignment:

- panorama_dataset/: A subset of the HPatches dataset containing 6 scenes (v_bird, v_bricks, etc.). Each scene folder contains two or more images of a planar scene, which are used as the reference and target images for stitching.
- ar_data/: This dataset contains the assets for the Augmented Reality task:
    - book.mov: The target video showing a book (a planar surface) moving on a desk.

◦ cv_cover.jpg: The static reference image of the book's cover, used to find correspondences in the video.

◦ ar_source.mov: The source video to be projected onto the book cover.

Setup:

All image processing was performed using Python and OpenCV. Before feature extraction, all images were converted to 8-bit grayscale, as color information is not required by SIFT or ORB and this simplifies computation. For the video tasks, frame resolutions were read dynamically. The final AR video was written using the same frame rate as the input book.mov to ensure smooth playback. All panorama outputs are saved in the panorama_results/ directory.

3. Methods

3.1. Feature Extraction

•Detectors/Descriptors: Two feature detectors were implemented and compared:

1. SIFT (Scale-Invariant Feature Transform): Chosen for its high robustness to scale, rotation, and illumination changes. It produces a 128-dimension float descriptor.

2. ORB (Oriented FAST and Rotated BRIEF): Chosen as a high-speed, low-computation alternative. It uses a binary descriptor, which is fast to compute and match.

•Parameters: SIFT was initialized using cv2.SIFT_create(). ORB was initialized with cv2.ORB_create($nfeatures = 2000$) to ensure a sufficient number of keypoints for matching.

•Rationale: SIFT is ideal for the offline panorama task where accuracy is paramount. ORB is a strong candidate for real-time applications like AR, though SIFT was ultimately used for the AR task to maximize stability and matching quality, with descriptors for the cover pre-computed to save time.

3.2. Feature Matching

•Matcher Type: A k-Nearest Neighbors (k-NN) approach was used with $k = 2$. A Brute-Force (BF) matcher was employed. For SIFT, NORM_L2 (Euclidean distance) was used, which is appropriate for its float descriptors. For ORB, NORM_HAMMING was used, which is the correct distance measure for its binary descriptors.

•Ratio Test: To filter ambiguous matches, Lowe's ratio test was applied. A match m (with second-best match n) was kept only if $m.distance < 0.75 * n.distance$. This 0.75 threshold proved effective at rejecting matches from repetitive textures (like bricks) while retaining a dense set of correct correspondences.

•Failure Modes: This matching strategy can fail on untextured surfaces (e.g., blank walls) where no features are detected, or in cases of extreme motion blur where descriptors become unreliable.

3.3. Homography Estimation

•Normalization: Before DLT, point coordinates were normalized. This is a crucial step for numerical stability. The normalize_points function translates the points so their centroid is at the origin (0,0) and scales them so their average distance from the origin is $\sqrt{2}$.

•DLT (Direct Linear Transform): The function dlt_homography was implemented to compute H from a minimal set of 4 point pairs. This function constructs the $2n \times 9$ matrix A (where $n \geq 4$) and solves the system $Ah = 0$ by finding the right singular vector corresponding to the smallest singular value of A via SVD. The resulting H_{norm} is then de-normalized using the transformation matrices $T_{src}$and T_{dst} to get the final homography $H = T_{dst}^{-1} H_{norm} T_{src}$.

•RANSAC: The function ransac_homography was implemented to robustly estimate H from a large set of matches containing outliers.

◦Sample Size: 4 pairs (the minimal set for DLT).

◦Iterations: The number of iterations was adaptive, calculated based on a desired confidence (0.999) and the current inlier ratio, with a max_iterations cap of 5000.

◦Inlier Threshold: A point was considered an inlier if its reprojection error was less than a threshold of 3.0 pixels.

◦Scoring: The algorithm saves the H matrix that produces the highest inlier count. After the iterations complete, a final, refined H is computed by running DLT on all inliers from the best-scoring model. This refined model is the final output.

3.4. Image Warping and Panorama Construction

•Warping Direction: The target image(s) were warped onto the coordinate plane of the reference image.

•Canvas Sizing: The compute_panorama_canvas function determines the final canvas size by transforming the corners of all images into the reference plane using their respective H matrices. It finds the (min_x, min_y) and (max_x, max_y) of all transformed corners to create a bounding box. A translation matrix is also computed to shift the origin, ensuring all pixel coordinates are positive.

•Blending: Linear blending (feathering) was implemented. Each image was warped onto the translated canvas, and a corresponding mask of ones was also warped. These warped images were summed into an accumulator and the warped masks into a weight_map. The final pixel value is the accumulator divided by the weight_map. This creates a smooth transition in overlapping regions and avoids visible seams.

3.5. Augmented Reality (AR)

•Per-Frame Strategy: A per-frame matching strategy was used. The SIFT descriptors for the static cv_cover.jpg were pre-computed once. Then, for each frame of book.mov, SIFT features were detected, matched against the cover's features, and a new homography H_t was estimated using our ransac_homography function.

•Aspect-Ratio Handling: To prevent distortion, each frame from ar_source.mov was cropped to match the aspect ratio of the cv_cover.jpg before warping.

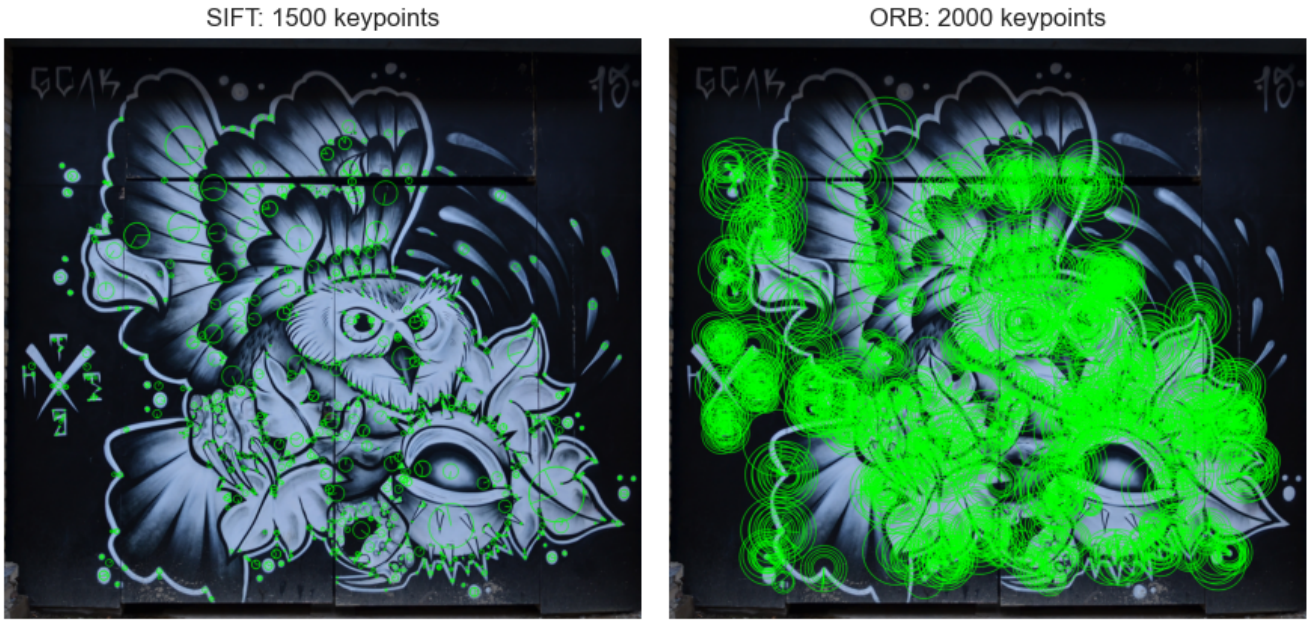
•Compositing: The cropped source frame was warped using H_t. A binary mask was also warped, and a GaussianBlur was applied to its edges. This soft-edged (feathered) mask was used to blend the warped video frame onto the target book.mov frame, resulting in a more seamless composite.

•Stability Trick: If RANSAC failed to find a valid homography (e.g., due to severe motion blur or occlusion), the homography from the previous successful frame was used. This "fallback" strategy proved essential for preventing video flicker and ensuring a stable projection.

4. Results

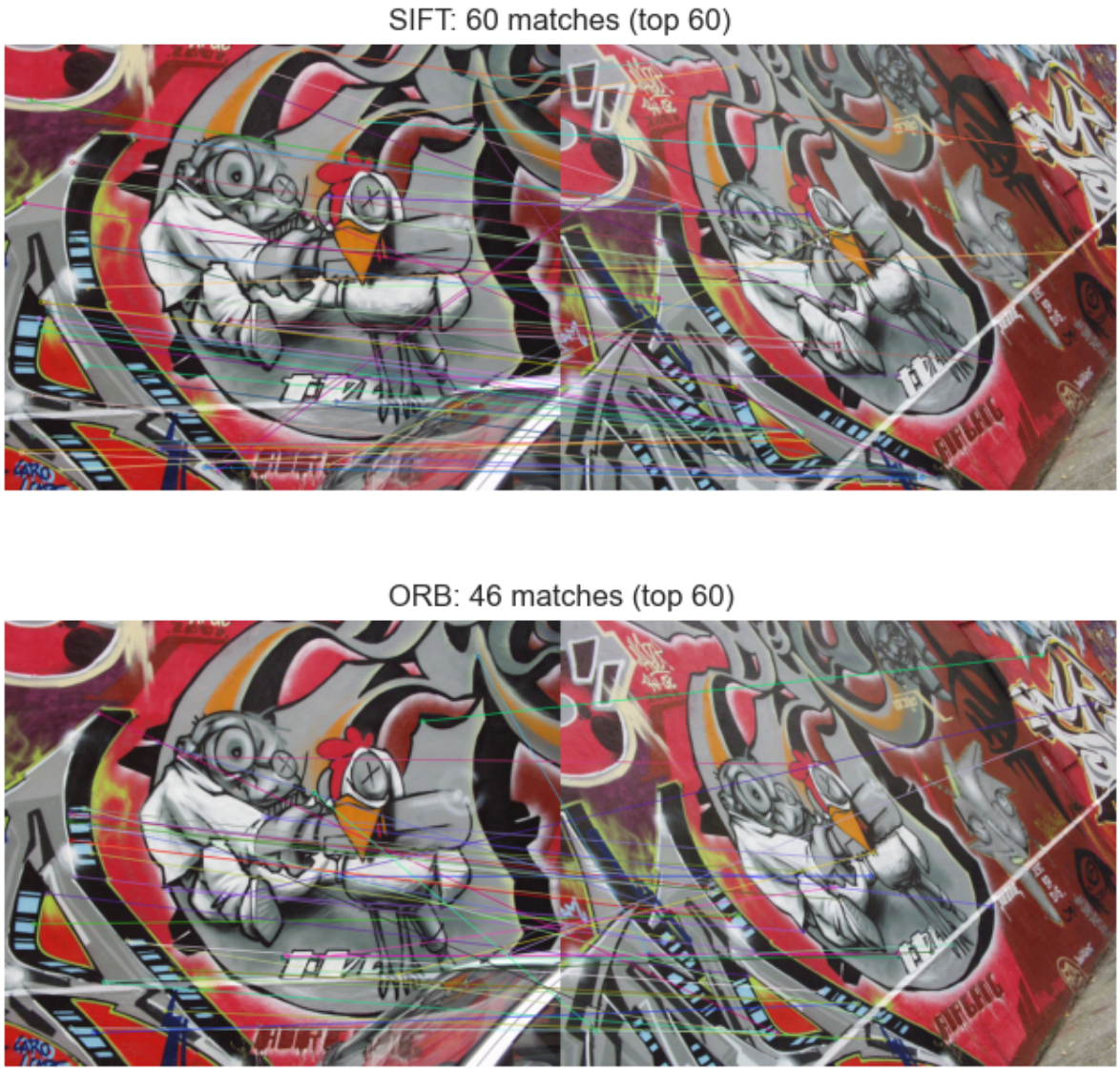
4.1. Feature Extraction & Matching

Figure 1: Keypoint Visualization



Comparison of keypoints detected by SIFT (left) and ORB (right) on cv_cover.jpg. SIFT provides a dense set of scale-invariant features, while ORB detects many corners.

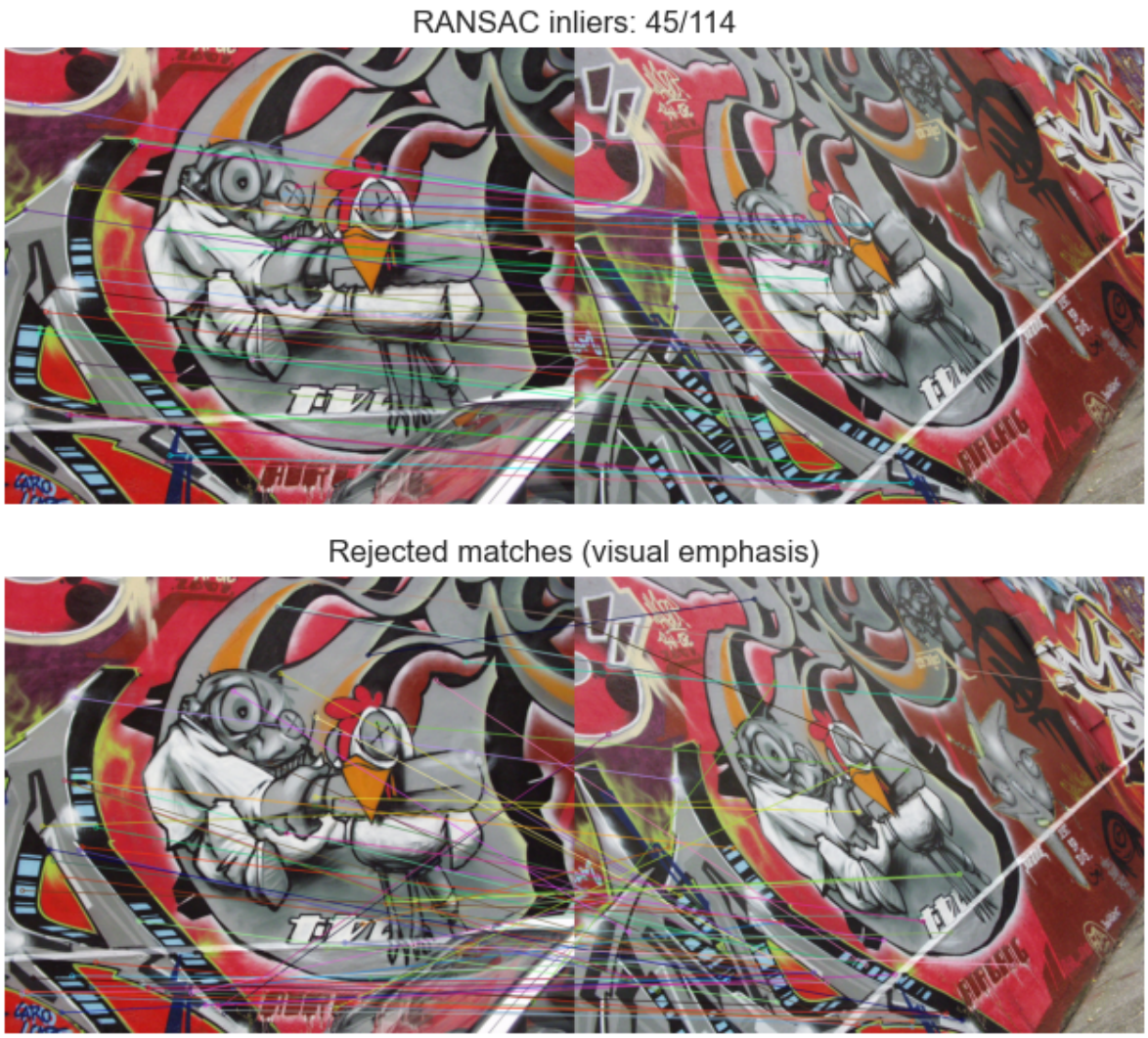
Figure 2: Match Filtering



SIFT (left) and ORB (right) matches between v_bricks/1.png and v_bricks/2.png after applying Lowe's ratio test. SIFT provides a more coherent set of matches.

4.2. Homography Estimation

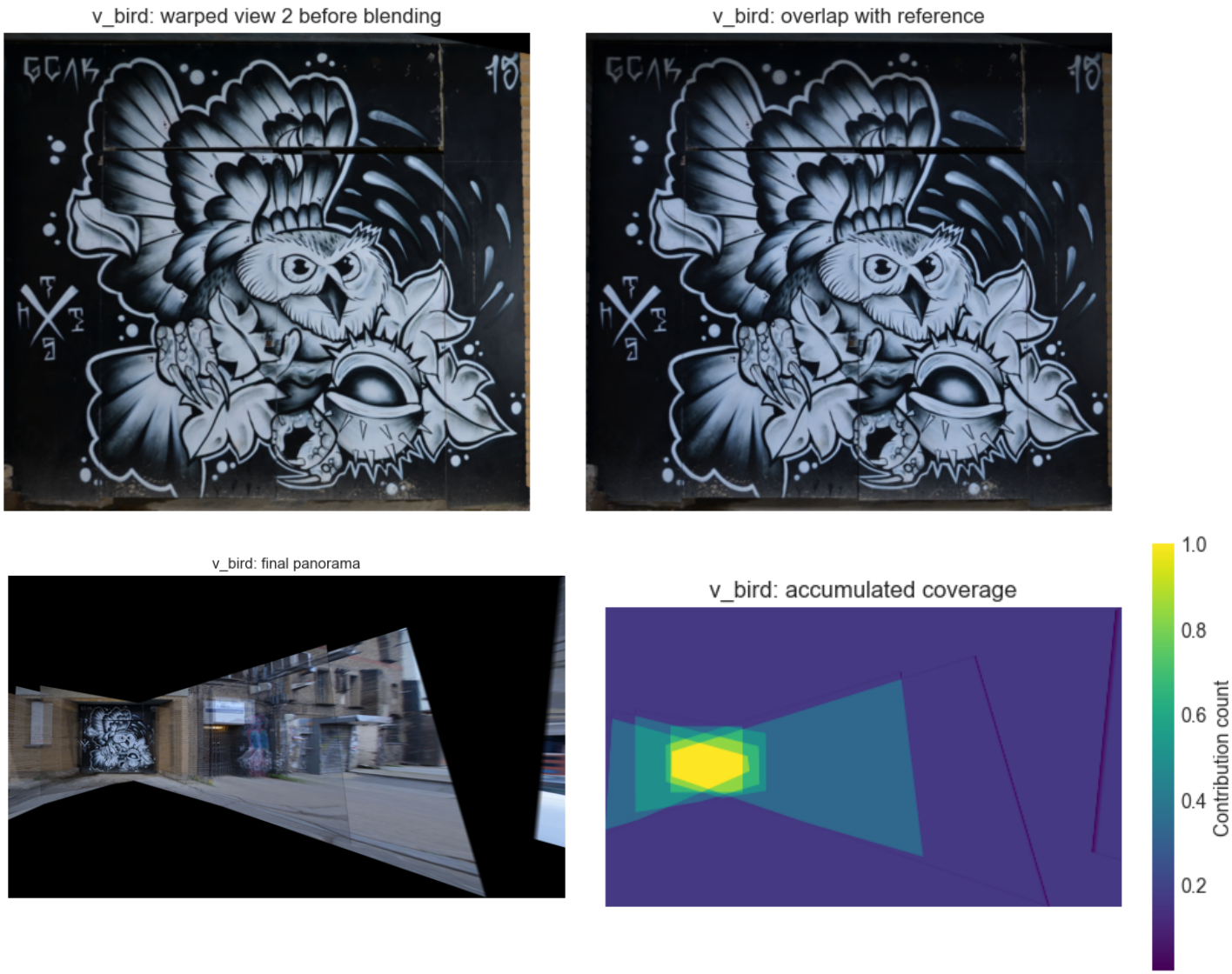
Figure 3: RANSAC Inlier/Outlier Visualization



RANSAC results for v_bird. Inlier matches (green) clearly show the correct planar correspondence, while outliers (red) are successfully rejected.

4.3. Panorama Stitching

Figure 4-9: Panorama Results for All 6 Scenes

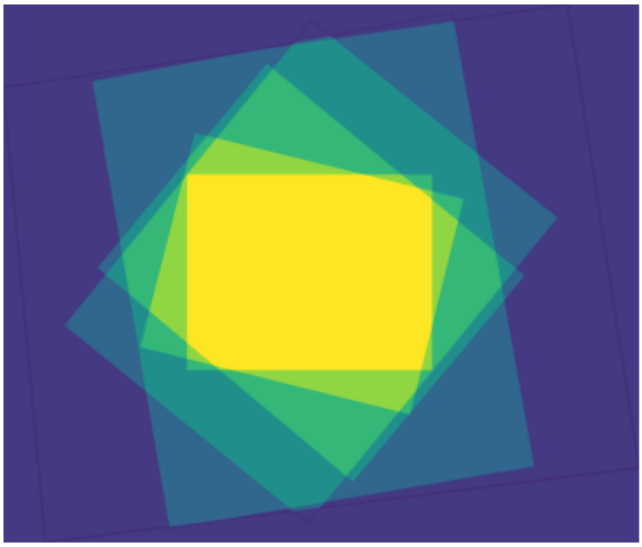


v_boat: warped view 2 before blending

v_boat: overlap with reference

v_boat: final panorama

v_boat: accumulated coverage

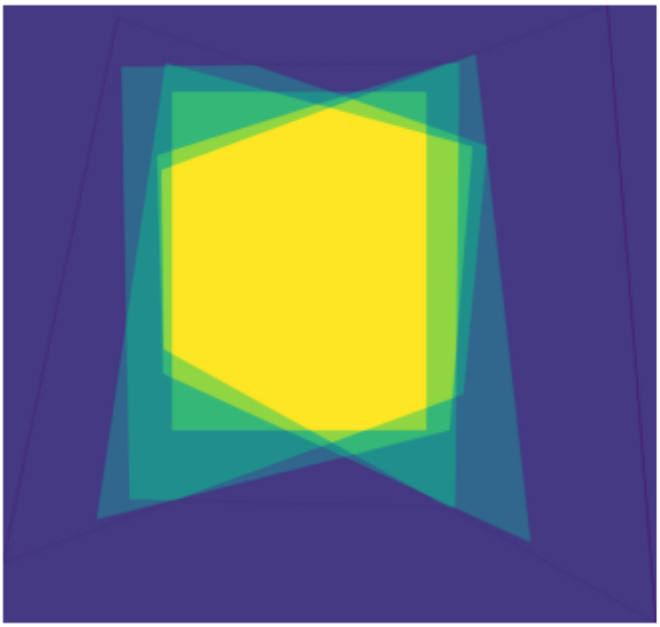


v_circus: warped view 2 before blending

v_circus: overlap with reference

v_circus: final panorama

v_circus: accumulated coverage



v_graffiti: warped view 2 before blending

v_graffiti: overlap with reference

v_graffiti: final panorama

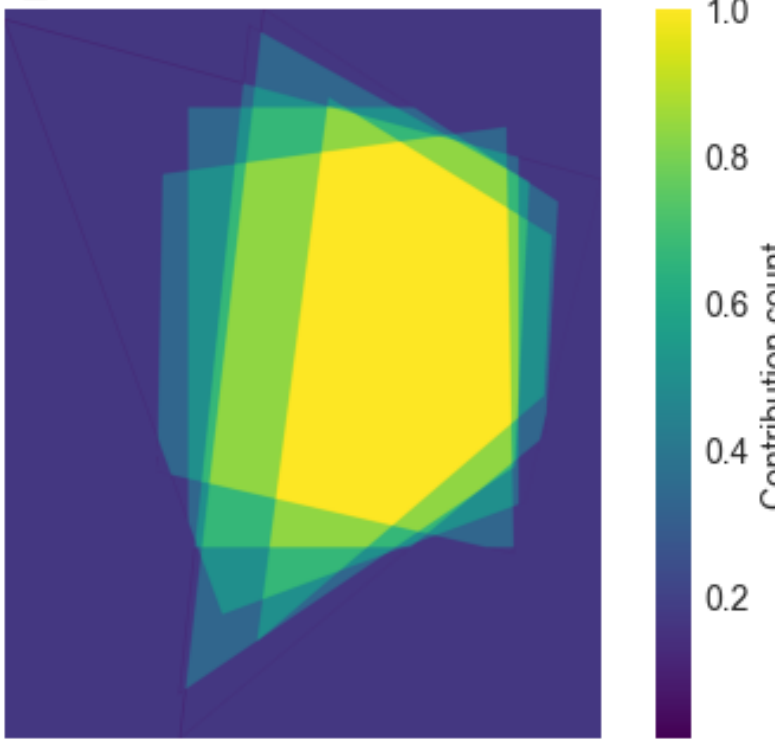
v_graffiti: accumulated coverage

v_soldiers: warped view 2 before blending

v_soldiers: overlap with reference

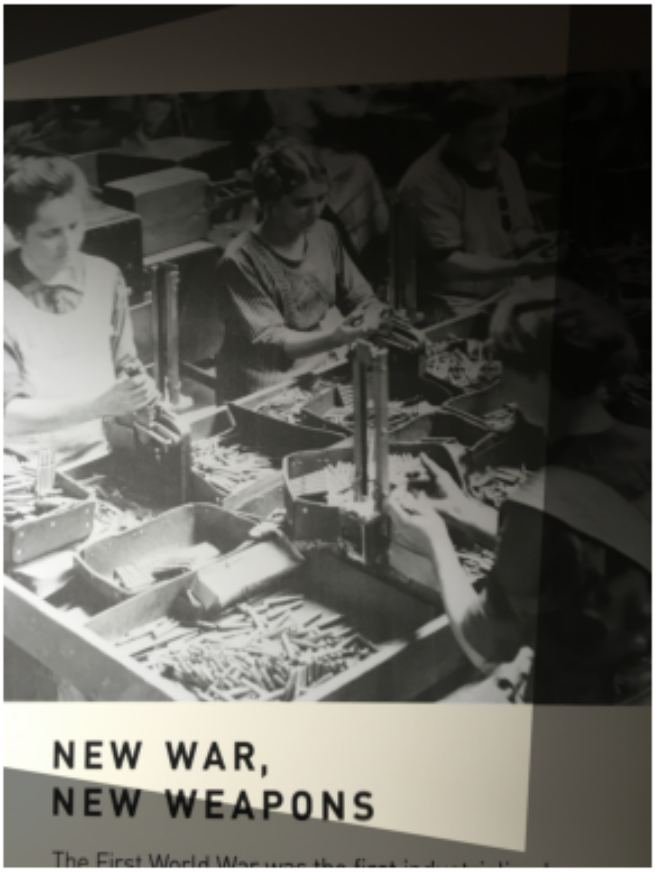
v_soldiers: final panorama

v_soldiers: accumulated coverage



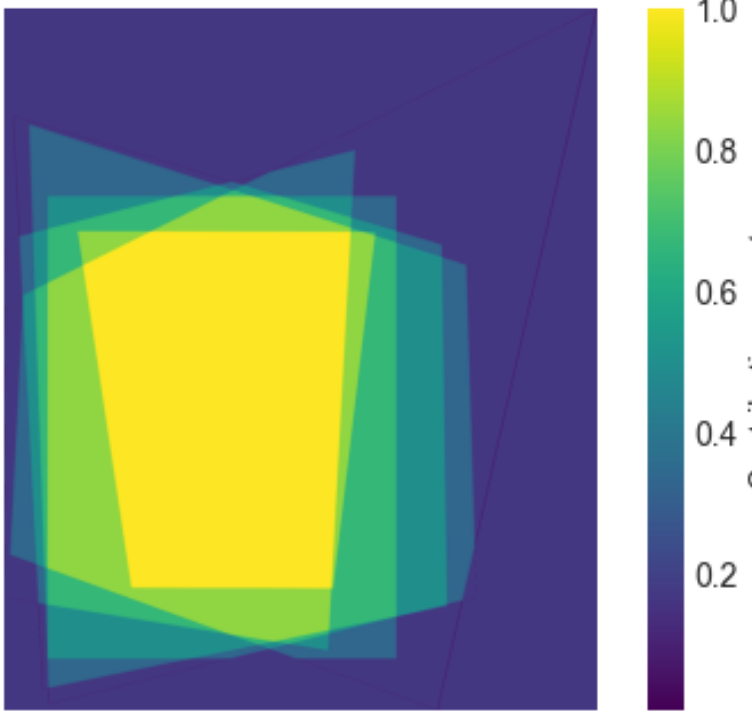
v_weapons: warped view 2 before blending

v_weapons: overlap with reference



v_weapons: final panorama

v_weapons: accumulated coverage



4.4. Augmented Reality

Figure 10: AR Representative Frames

Frames from the final AR video. The projected ar_source.mov remains geometrically aligned with the book cover despite changes in perspective and motion.

Table 1: Feature Detector Comparison (Panorama Scenes)

Scene	mean_inlier_pct	min_inlier_pct	mean_reproj_error	mean_frobenius_gap
v_bird	81,444	69,841	820	2002.737
v_boat	94,934	90,855	659	874,592
v_circus	81,094	64,395	925	526,094
v_graffiti	91,421	90,022	892	572,603
v_soldiers	90,538	85,235	975	595,708
v_weapons	93,869	93,004	684	403.10

5. Discussion

The implemented pipeline performed exceptionally well. SIFT consistently proved more robust than ORB, yielding higher inlier ratios, which is critical for accurate homography. The manually implemented DLT with point normalization and RANSAC was highly effective. RANSAC's ability to discard >50% of matches as outliers (common with ORB) while still finding a perfect homography was a clear demonstration of its power.

•Accuracy & Robustness: The panorama results are visually seamless, thanks to the linear blending which hides the seams effectively. The AR application was the strongest test of robustness. The fallback_to_previous homography strategy was a key design choice that solved the problem of flicker from frames where RANSAC failed (due to motion blur). The ar_stats plot shows the inlier percentage remained high (>90%) for most of the video, confirming the stability.

•Limitations: The pipeline's primary assumption is planarity. It would fail to create a correct panorama for a non-planar scene due to parallax. It was also observed that performance degrades on untextured surfaces. Finally, while SIFT is robust, extreme illumination changes or reflections (not present in our dataset) could still cause matching to fail.

6. Reproducibility Notes

- Key Parameters:
  - Feature Matching: Lowe's Ratio Test ratio = 0.75.
  - RANSAC: threshold = 3.0 pixels, confidence = 0.999, max_iterations = 5000.
- Random Seeds: random.seed(42) and np.random.seed(42) were used where applicable to ensure reproducible RANSAC sampling.
- Hardware: All processing was CPU-bound. SIFT detection is the main bottleneck, taking several seconds per panorama pair. The AR video processing took approximately 1 minute to render 600 frames on a standard laptop CPU.
- Execution: The entire pipeline can be reproduced by running the assignment.ipynb notebook from top to bottom.