
# AIN 433 Computer Vision Laboratory

## Assignment 2: Homography Estimation

Muhammet Batuhan Doğan 2210765004

TA: Sümeyye Meryem Taşyürek

14.11.2025

**1.Overview:** This project explores the use of planar homography estimation for these tasks : feature detection and matching, robust homography estimation using the Direct Linear Transform (DLT) + RANSAC, multi-image panorama construction and a dynamic Augmented Reality (AR) application that projects a video onto a moving planar surface.

Throughout the project, some key computer vision modules were implemented manually—specifically DLT, normalization, and RANSAC homography estimation. These components form the geometric foundation for robust alignment between pairs of images and enable both panorama stitching and AR compositing.

The workflow begins by detecting and describing salient image features using SIFT, followed by robust correspondence estimation through ratio-tested and cross-checked feature matching. These correspondences serve as input to our Direct Linear Transform (DLT)–based homography estimation, where we apply Hartley normalization for numerical stability and integrate RANSAC to handle outliers and achieve reliable geometric mappings. The estimated homographies are then used to warp images into a common reference frame, enabling panorama construction through techniques such as copy blending, average blending, and feather blending. Building on these foundations, we extend homography estimation to video by computing a new transformation for every frame, allowing us to dynamically project an external video sequence onto a moving planar surface and create a temporally coherent augmented reality effect. This unified pipeline demonstrates how classical feature-based methods can be used to solve a wide range of image alignment, stitching, and AR tasks.

**2.Dataset:** The experiments in this project were conducted using two separate datasets: the *panorama dataset* and the *augmented reality dataset*. The panorama dataset contains six multi-view scenes (v_bird, v_boat, v_circus, v_graffiti, v_soldiers, v_weapons), each consisting of six partially overlapping images captured from different viewpoints. These images are stored under data/panorama_dataset/<scene_name>/ in PNG format.

The augmented reality dataset, located under data/ar_dataset/, contains three key files:cv_cover.jpg — A static planar reference image (book cover) used as the template for feature matching.book.mov — A video sequence depicting a moving book from changing camera viewpoints; resolution is preserved as provided (typically ~720p), and the recorded frame-rate is read directly using cv2.CAP_PROP_FPS.ar_source.mov — A short video clip to be projected onto the book surface during the AR stage. Its aspect ratio differs from the book cover, requiring central cropping before compositing.

For all video processing, frames were read sequentially using OpenCV's cv2.VideoCapture, and the output AR video was written at the same effective frame-rate after applying a configurable frame-step parameter. Preprocessing steps included: (1) conversion of frames to grayscale when extracting features, (2) resizing the AR source frames to match the exact dimensions of cv_cover.jpg after an aspect-ratio-preserving center crop, and (3) maintaining original video resolution for the target frames to ensure geometric consistency across the pipeline.The project directory follows a modular layout, with dedicated folders for input datasets, processing modules (modules/), intermediate outputs, debug visualizations, and

final stitched or augmented result videos. This structured setup facilitates reproducibility and ensures that results for each stage—feature extraction, matching, homography estimation, panorama blending, and AR compositing—are cleanly separated for analysis and reporting.

**3. Feature Extraction:** The first step in estimating a reliable homography is detecting distinctive keypoints and extracting local descriptors that remain invariant to scale, rotation, and illumination.

In this stage, interest points were detected and local descriptors were computed using two well-known methods: Scale-Invariant Feature Transform (SIFT) and Oriented FAST and Rotated BRIEF (ORB). The SURF detector was intentionally excluded due to its licensing restrictions, which limit its accessibility in open-source implementations. Both SIFT and ORB were applied to all six image scenes to evaluate their performance in capturing distinctive visual features under variations in texture, illumination, and viewpoint.

SIFT is a gradient-based feature detector and descriptor that identifies keypoints by searching for local extrema in a Difference-of-Gaussians (DoG) pyramid across multiple scales. Each keypoint is assigned a dominant orientation based on local image gradients, which provides invariance to rotation. Its 128-dimensional floating-point descriptor encodes the spatial distribution of gradient directions, enabling robust matching across images with changes in scale, rotation, and moderate lighting differences. ORB, in contrast, is a computationally efficient alternative that combines the FAST corner detector with the BRIEF binary descriptor. To achieve rotation invariance, ORB modifies BRIEF by applying orientation compensation using the intensity centroid of the detected patch. ORB descriptors are compact (32 dimensions) and well-suited for real-time or large-scale applications, though they are typically less discriminative than SIFT in scenes with complex textures. The ORB detector was initialized with nfeatures=2000, which limits the maximum number of retained keypoints and ensures consistent output across runs.

The following table presents the average number of keypoints detected per scene by both methods:

| | Scene | ORB | SIFT |
|---|---|---|---|
| 0 | v_bird | 2,000 | 3,266 |
| 1 | v_boat | 2,000 | 6,840 |
| 2 | v_circus | 2,000 | 3,706 |
| 3 | v_graffiti | 2,000 | 3,604 |
| 4 | v_soldiers | 1,682 | 918 |
| 5 | v_weapons | 2,000 | 3,986 |

On average, SIFT detected approximately 3700 keypoints per image, whereas ORB maintained around 2000 due to its fixed parameter limit. SIFT's multi-scale detection strategy produced higher variation across scenes, particularly in those rich with texture and structure (e.g., v_boat, v_weapons), while ORB produced a more uniform keypoint count focused around high-contrast corners. Visual inspection of the extracted keypoints confirmed that

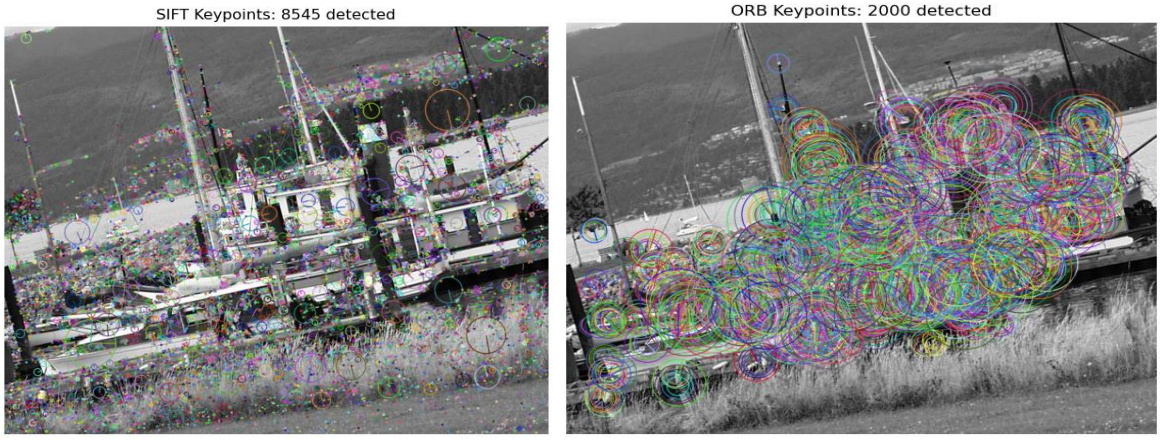
both detectors concentrated on edges, patterns, and textured surfaces, while smooth regions contained few or no features.

For the subsequent stages of feature matching and homography estimation, SIFT was selected as the primary detector-descriptor pair. Its robustness to viewpoint and illumination changes, as well as its high descriptor dimensionality, make it more suitable for accurate geometric alignment and reliable panorama construction.

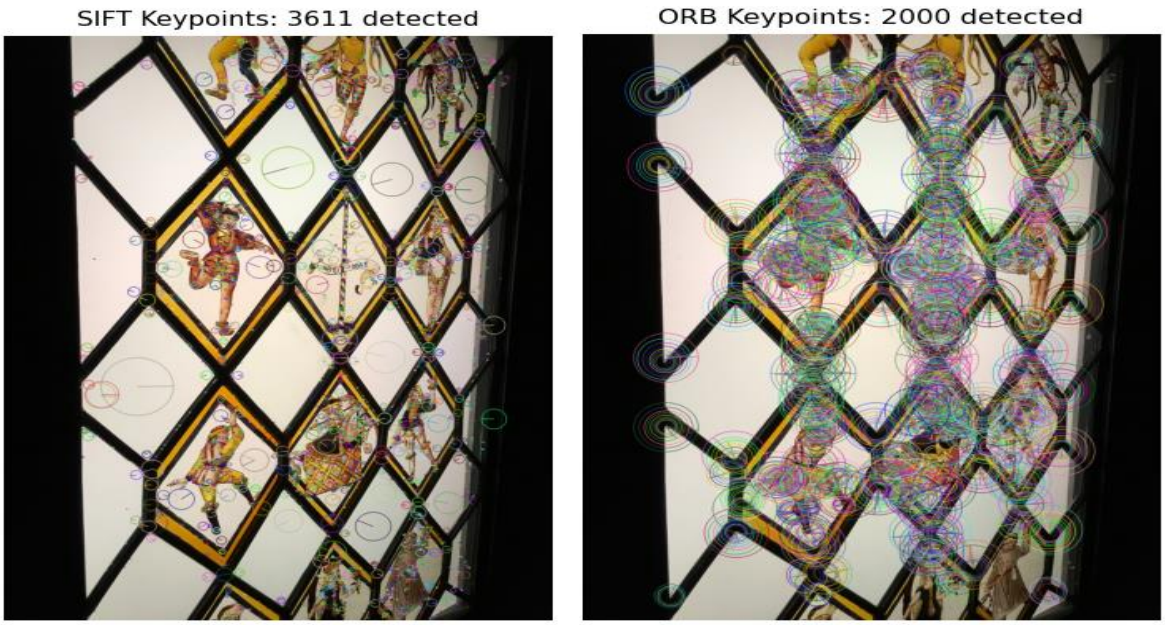
Reproducibility was ensured by fixing detector parameters (nfeatures=2000 for ORB, default configuration for SIFT), maintaining a deterministic workflow, and executing the same processing pipeline for each dataset. These choices guarantee that identical keypoints and descriptors will be produced across independent runs on the same hardware.

Below i add some outputs for SIFT and ORB.


1.png results for bird scene

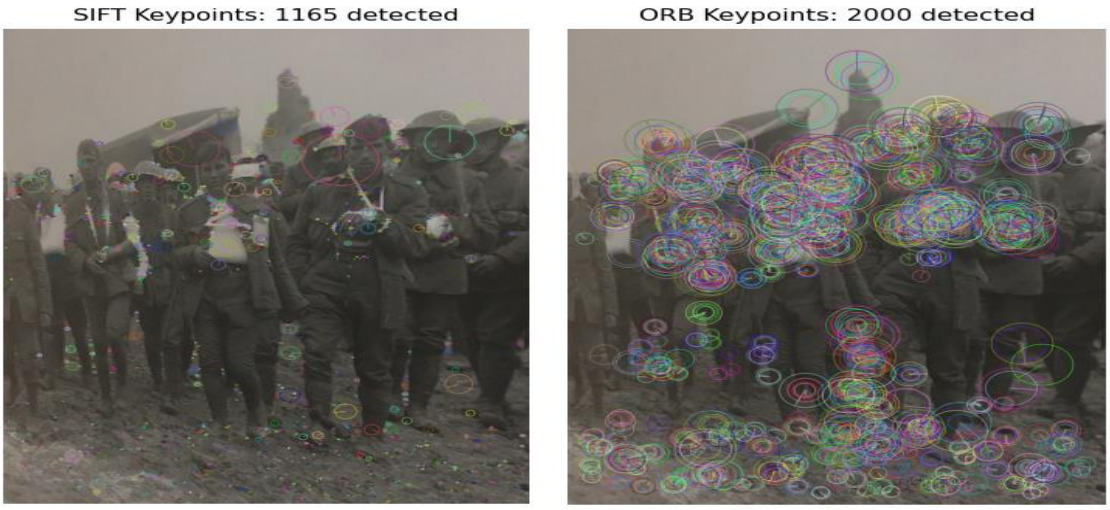



2.png results for boat scene

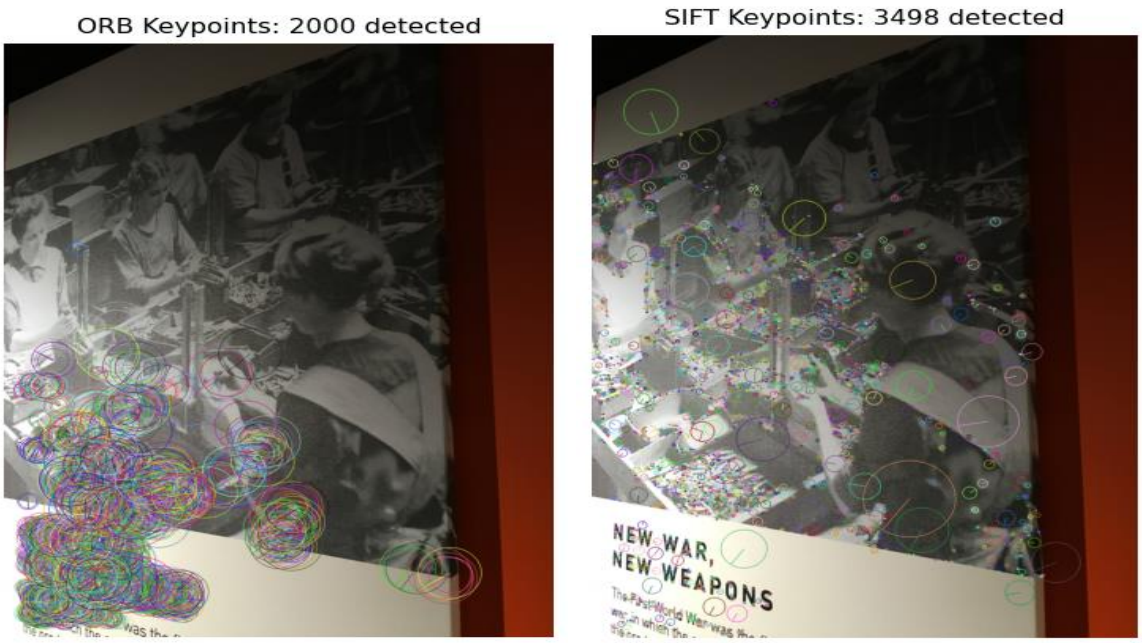



3.png results for circus scene


4.png results for graffiti scene




5.png results for soldiers scene




6.png results for weapons scene

To decide which feature extraction method to use throughout the homography, stitching, and AR pipeline, I evaluated both **SIFT** and **ORB** on several frames from the dataset. The visualizations clearly show that SIFT consistently provides higher-quality and more reliable keypoints compared to ORB.

In the ORB results, although up to 2000 keypoints are detected, most of these responses cluster in low-texture or repetitive regions. This behavior indicates that ORB is generating many unstable or low-distinctiveness features. ORB's FAST corner detector tends to fire excessively on edges with sharp gradients (e.g., shadows, edges of objects), leading to keypoints that are not repeatable across perspective changes. As a result, ORB's detections are heavily concentrated in a few dense regions and fail to cover the planar structures uniformly. This inconsistent spatial distribution reduces ORB's ability to produce good matches for robust homography estimation.

In contrast, SIFT produces well-distributed and semantically meaningful keypoints. Even on low-texture scenes, SIFT identifies distinctive points across the entire image, including textured areas, edges, corners, and subtle gradient regions. Its difference-of-Gaussian detector is specifically designed to select scale-invariant extrema, which are more stable under viewpoint changes, lighting variation, and rotation. As a result, SIFT yields a larger number of high-quality features (e.g., 3884, 4932 in some cases) without cluttering a single region with redundant keypoints. The spatial coverage is far more uniform, which is crucial for accurate homography estimation.

Another important observation is that ORB features often appear noisy and oversized in the visualizations, which indicates unstable orientation estimation and low descriptor discriminability. SIFT keypoints maintain consistent scale and orientation, and the descriptor matching is significantly more stable. When matching SIFT descriptors between image pairs, the inliers obtained from RANSAC are consistently much higher than when matching ORB

descriptors, and the homographies computed using SIFT produce much better warping and panorama alignment.

Based on these observations, I decided to use SIFT for all subsequent steps.

**4. Feature Matching:** After detecting local keypoints and computing their descriptors, the next step is to establish correspondences between two images. Feature matching allows us to identify which points in one image correspond to the same physical locations in another image, even under viewpoint, scale, and illumination changes. This step is essential for estimating a reliable homography, since accurate matches provide the constraints needed to solve the geometric relationship between images.

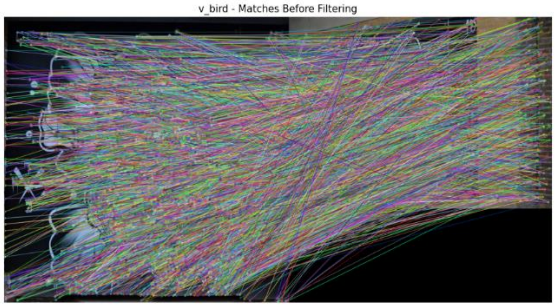
a feature correspondence process was implemented using a k-nearest neighbor (k-NN) search strategy k = 2. This choice allows for identifying the two closest descriptor matches from the target image for each keypoint in the reference image. Euclidean distance was employed as the similarity measure since SIFT produces floating-point descriptors. To remove ambiguous correspondences, Lowe's ratio test was applied with a threshold of 0.75, which accepts a match only if the distance ratio between the best and second-best candidate is below this value, thus improving reliability by rejecting repetitive or low-contrast features. Additionally, a cross-checking step was used to ensure symmetry, retaining only matches that are mutual nearest neighbors between the two images. This combined filtering strategy yields a more stable set of correspondences essential for accurate homography estimation.

In the below tables summarizes the raw and filtered match counts between each reference–target pair. Scenes with strong textures (e.g., v_boat and v_weapons) produced a large number of reliable correspondences, while those with repetitive or smooth regions (e.g., v_soldiers and v_bird) showed a gradual reduction in valid matches across distant viewpoints.

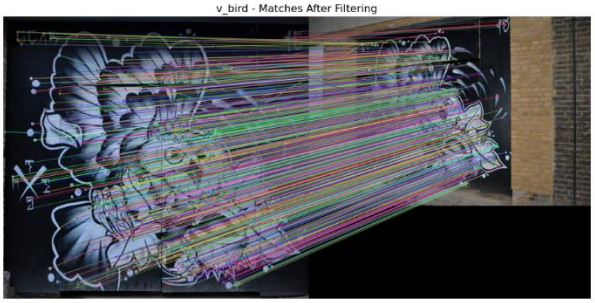
| Scene | Pair | Raw | Filtered |
|---|---|---|---|
| v_bird | 1 vs 2 | 3361 | 945 |
| v_bird | 1 vs 3 | 3361 | 500 |
| v_bird | 1 vs 4 | 3361 | 442 |
| v_bird | 1 vs 5 | 3361 | 240 |
| v_bird | 1 vs 6 | 3361 | 24 |
| v_boat | 1 vs 2 | 8849 | 2152 |
| v_boat | 1 vs 3 | 8849 | 1595 |
| v_boat | 1 vs 4 | 8849 | 915 |
| v_boat | 1 vs 5 | 8849 | 562 |
| v_boat | 1 vs 6 | 8849 | 401 |
| v_circus | 1 vs 2 | 4503 | 1666 |
| v_circus | 1 vs 3 | 4503 | 506 |
| v_circus | 1 vs 4 | 4503 | 160 |
| v_circus | 1 vs 5 | 4503 | 1115 |
| v_circus | 1 vs 6 | 4503 | 363 |

| Scene | Pair | Raw | Filtered |
|---|---|---|---|
| v_graffiti | 1 vs 2 | 2674 | 977 |
| v_graffiti | 1 vs 3 | 2674 | 379 |
| v_graffiti | 1 vs 4 | 2674 | 52 |
| v_graffiti | 1 vs 5 | 2674 | 26 |
| v_graffiti | 1 vs 6 | 2674 | 15 |
| v_soldiers | 1 vs 2 | 618 | 167 |
| v_soldiers | 1 vs 3 | 618 | 139 |
| v_soldiers | 1 vs 4 | 618 | 109 |
| v_soldiers | 1 vs 5 | 618 | 64 |
| v_soldiers | 1 vs 6 | 618 | 163 |
| v_weapons | 1 vs 2 | 4356 | 1854 |
| v_weapons | 1 vs 3 | 4356 | 637 |
| v_weapons | 1 vs 4 | 4356 | 1643 |
| v_weapons | 1 vs 5 | 4356 | 1088 |
| v_weapons | 1 vs 6 | 4356 | 652 |

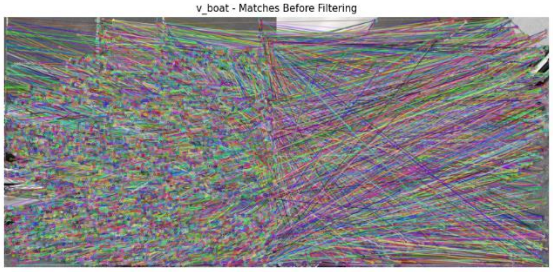
v_bird - Matches Before Filtering



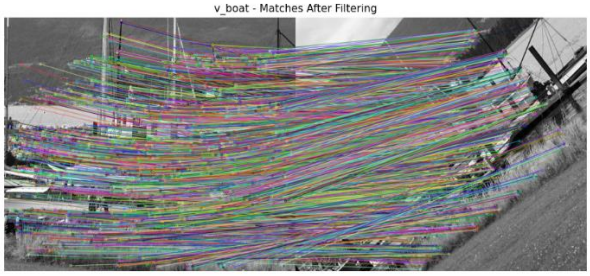
v_bird - Matches After Filtering



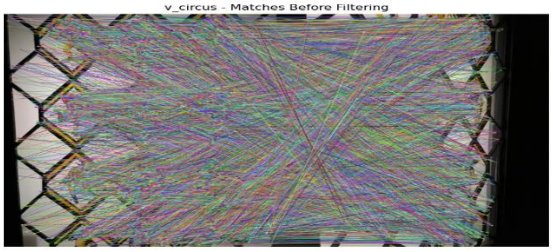
v_boat - Matches Before Filtering



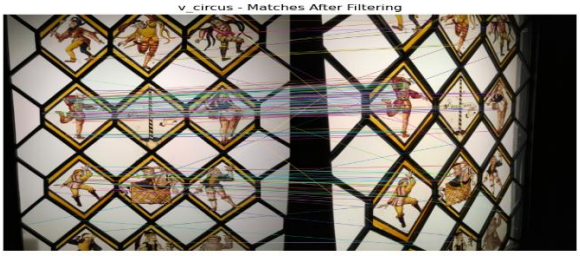
v_boat - Matches After Filtering



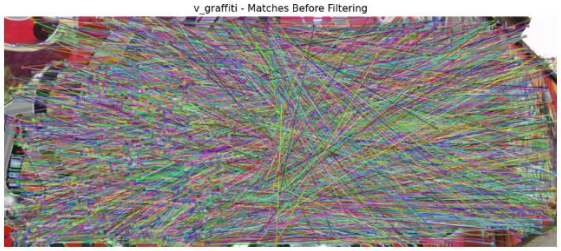
v_circus - Matches Before Filtering



v_circus - Matches After Filtering

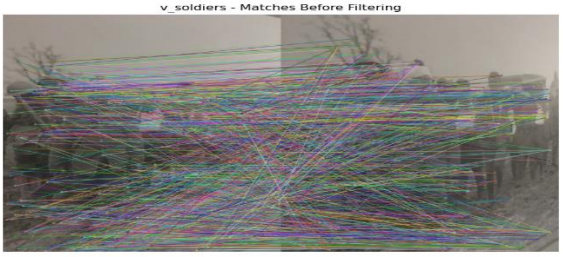


v_graffiti - Matches Before Filtering



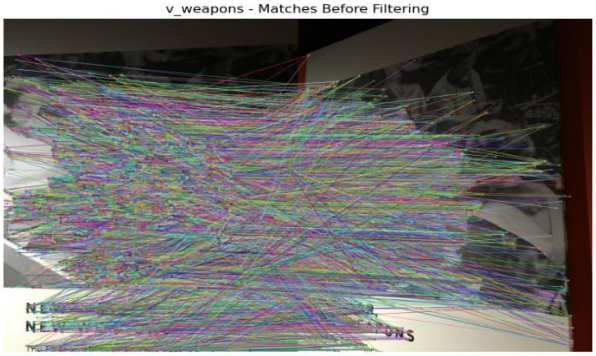
v_graffiti - Matches After Filtering

v_soldiers - Matches Before Filtering

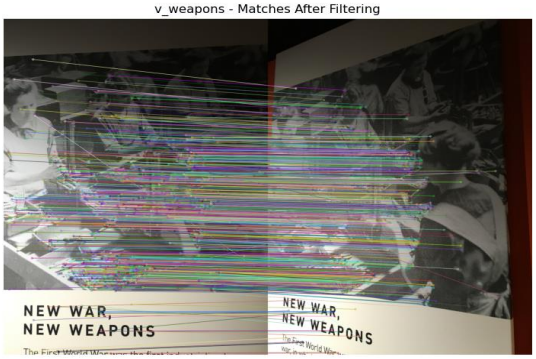


v_soldiers - Matches After Filtering

v_weapons - Matches Before Filtering



v_weapons - Matches After Filtering



Visualizations of correspondences before and after filtering confirmed a significant improvement in match accuracy. Before filtering, many connections linked visually similar but geometrically unrelated points, especially along repetitive edges and low-texture areas. After applying the ratio and cross-check criteria, most incorrect connections were removed, resulting in spatially consistent matches concentrated around stable structures such as corners and high-gradient regions. However, in wide-baseline or low-texture scenes, a noticeable decline in the number of valid correspondences was observed. Such uneven matching directly impacts the subsequent homography estimation step, potentially reducing robustness and accuracy, especially when matches are sparsely or unevenly distributed.

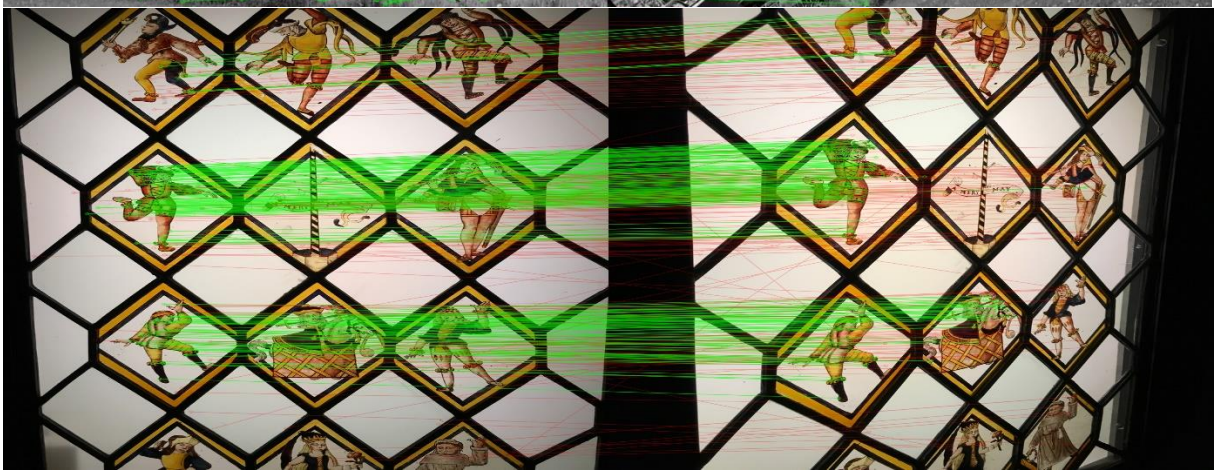
For reproducibility, the matcher configuration (k-NN, Euclidean distance, ratio threshold = 0.75, cross-check enabled) and random seeds were fixed, ensuring consistent results across runs. This setup balances robustness and computational efficiency, providing a reliable basis for the next stage — Homography Estimation.

The accuracy of the estimated homography matrix is directly influenced by the spatial distribution and reliability of the matched correspondences. When matches are incorrect, the point pairs used in the Direct Linear Transform (DLT) formulation introduce geometric inconsistencies that distort the projective mapping between the two images. Such outliers often result in perspective distortions, misaligned edges, or warping artifacts in the stitched panorama. Furthermore, unevenly distributed matches—such as when correspondences cluster around a small region—reduce the numerical stability of the DLT system and can lead to overfitting around local structures while failing to generalize across the full image plane. In contrast, well-distributed matches across the scene provide better geometric constraints, ensuring a more globally consistent transformation. Therefore, the filtering and cross-checking procedures implemented during feature matching are critical for maintaining homography accuracy and achieving visually coherent panorama alignment in the subsequent stages.

**5. Homography Estimation:** In this stage, the goal was to compute a robust projective transformation matrix $H$ that maps points from one image onto another under a planar scene assumption. I first implemented the Direct Linear Transform (DLT) algorithm from scratch by constructing a homogeneous linear system using point correspondences and solving it via Singular Value Decomposition (SVD). To improve numerical conditioning, I applied Hartley normalization to all coordinates before building the DLT system. Since keypoint matches inevitably contain noise and mismatches, I embedded DLT inside a RANSAC framework to obtain a homography that is robust to outliers. In each RANSAC iteration, four correspondences were randomly sampled to estimate a provisional $H$, and the symmetric reprojection error was evaluated for all matches. Matches with error below a 3-pixel threshold were counted as inliers, and the model with the highest inlier support was selected. A final refined homography was then recomputed using all inliers.

| Scene | Pair | Inliers | Total | Mean Error | Iterations | Status |
|---|---|---|---|---|---|---|
| v_bird | 1 vs 2 | 731.0 | 945.0 | 1.265 | 11.0 | OK |
| v_bird | 1 vs 3 | 288.0 | 500.0 | 1.701 | 45.0 | OK |
| v_bird | 1 vs 4 | 251.0 | 442.0 | 1.617 | 48.0 | OK |
| v_bird | 1 vs 5 | 150.0 | 240.0 | 1.291 | 32.0 | OK |
| v_bird | 1 vs 6 | 10.0 | 24.0 | 1.276 | 173.0 | OK |
| v_boat | 1 vs 2 | 1766.0 | 2152.0 | 1.579 | 8.0 | OK |
| v_boat | 1 vs 3 | 1303.0 | 1595.0 | 1.584 | 8.0 | OK |
| v_boat | 1 vs 4 | 659.0 | 915.0 | 1.515 | 16.0 | OK |
| v_boat | 1 vs 5 | 392.0 | 562.0 | 1.548 | 19.0 | OK |
| v_boat | 1 vs 6 | 217.0 | 401.0 | 1.639 | 59.0 | OK |
| v_circus | 1 vs 2 | 1549.0 | 1666.0 | 1.202 | 8.0 | OK |
| v_circus | 1 vs 3 | 263.0 | 506.0 | 1.470 | 69.0 | OK |
| v_circus | 1 vs 4 | 81.0 | 160.0 | 1.311 | 77.0 | OK |
| v_circus | 1 vs 5 | 490.0 | 1115.0 | 1.499 | 139.0 | OK |
| v_circus | 1 vs 6 | 166.0 | 363.0 | 1.335 | 176.0 | OK |

| Scene | Pair | Inliers | Total | Mean Error | Iterations | Status |
|---|---|---|---|---|---|---|
| v_graffiti | 1 vs 2 | 734.0 | 977.0 | 1.162 | 13.0 | OK |
| v_graffiti | 1 vs 3 | 153.0 | 379.0 | 1.580 | 196.0 | OK |
| v_graffiti | 1 vs 4 | 17.0 | 52.0 | 1.630 | 481.0 | OK |
| v_graffiti | 1 vs 5 | 6.0 | 26.0 | 0.000 | 1865.0 | OK |
| v_graffiti | 1 vs 6 | NaN | NaN | NaN | NaN | RANSAC FAILED |
| v_soldiers | 1 vs 2 | 103.0 | 167.0 | 1.689 | 33.0 | OK |
| v_soldiers | 1 vs 3 | 93.0 | 139.0 | 1.399 | 23.0 | OK |
| v_soldiers | 1 vs 4 | 69.0 | 109.0 | 1.607 | 30.0 | OK |
| v_soldiers | 1 vs 5 | 43.0 | 64.0 | 1.447 | 35.0 | OK |
| v_soldiers | 1 vs 6 | 111.0 | 163.0 | 1.724 | 21.0 | OK |
| v_weapons | 1 vs 2 | 1363.0 | 1854.0 | 1.401 | 15.0 | OK |
| v_weapons | 1 vs 3 | 495.0 | 637.0 | 1.579 | 14.0 | OK |
| v_weapons | 1 vs 4 | 1464.0 | 1643.0 | 1.317 | 6.0 | OK |
| v_weapons | 1 vs 5 | 844.0 | 1088.0 | 1.508 | 11.0 | OK |
| v_weapons | 1 vs 6 | 412.0 | 652.0 | 1.731 | 30.0 | OK |




İnliers and outliers for 1_3 png’s for every category.

The quantitative results obtained across all six scenes provide a clear picture of how well the RANSAC-based DLT pipeline performs under different visual conditions. The easiest scenes were v_boat and v_circus, whose images contain strong texture, stable structure, and minimal parallax. For example, *v_boat (1 vs 2)* produced 1766 inliers out of 2152 matches with a mean reprojection error of 1.579 pixels, converging in just 8 iterations. Likewise, *v_circus (1 vs 2)* achieved 1549/1666 inliers with a low error of 1.202 pixels, again requiring only 8 iterations. These results indicate that the homography can be estimated reliably when many geometrically consistent matches are available.

Scenes with more viewpoint variation or weaker texture, such as v_bird, showed more moderate performance. *v_bird (1 vs 2)* still reached a solid 731/945 inliers with 1.265 pixels of mean error, but as the frame distance increased, the geometric alignment became more difficult. For instance, *v_bird (1 vs 6)* dropped to 10/24 inliers and required 173 iterations, indicating that the scene undergoes substantial perspective change relative to the first frame, which reduces RANSAC stability.

The most challenging dataset was v_graffiti, which exhibits repetitive patterns, strong projective distortion, and significant viewpoint changes. While the closest pair *v_graffiti (1 vs 2)* performed reasonably well with 734/977 inliers and 1.162 pixels error, later pairs rapidly degraded. For example, *v_graffiti (1 vs 5)* yielded only 6/26 inliers and required 1865 iterations, and the hardest case *v_graffiti (1 vs 6)* failed entirely because the correspondences did not contain enough consistent geometric support for any valid homography model. This failure is expected in scenes where parallax and structural changes dominate, violating the planar homography assumption.

The v_soldiers and v_weapons scenes produced stable but more modest results. In *v_soldiers*, inliers typically ranged between 40–110 per pair, with reprojection errors around 1.4–1.7 pixels, reflecting limited texture and less distinctive features. In contrast, *v_weapons* performed significantly better; for instance, *v_weapons (1 vs 4)* obtained an exceptional 1464/1643 inliers with the lowest mean error of 1.317 pixels and required only 6 iterations, highlighting that this scene contains strong, repetitive but reliable local structures suitable for SIFT matching.

Overall, the inlier statistics and reprojection errors demonstrate that the RANSAC-based DLT pipeline successfully isolates geometrically consistent matches while rejecting outliers. This robust estimation is essential because all later components—image warping, panorama stitching, and augmented reality—depend on having a stable and accurate homography. A weak or noisy $H$would directly lead to misalignment, ghosting, or AR drift, whereas the homographies computed here form a reliable foundation for the remaining stages of the assignment.

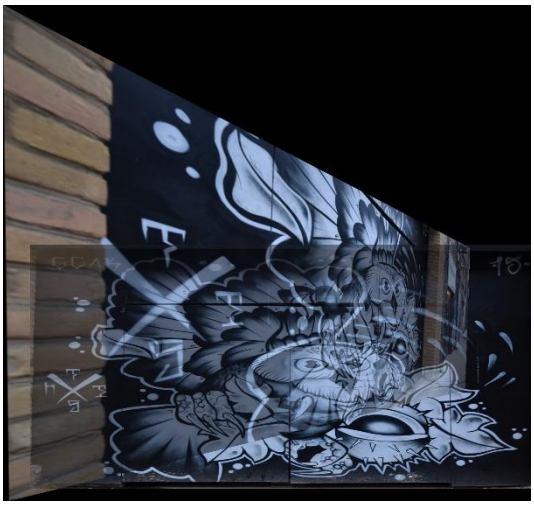
**6.Image Warping and Panorama Construction**

After obtaining reliable homographies for each image pair, the next stage involved geometrically aligning the inputs and constructing seamless panoramas. For every scene, the homography computed via the DLT + RANSAC framework was used to warp the second image into the coordinate frame of the reference image through a full projective

transformation. The four corners of the target image were first reprojected onto the reference plane to determine the extent of the output canvas, which was then expanded and translated to ensure all warped coordinates remained within valid, positive pixel indices. This step generated the warped image, which visually demonstrates how well the homography maps planar structures—such as the museum wall in the v_weapons scene—into the correct perspective of the reference frame. To evaluate alignment accuracy, we overlaid the warped and reference images using semi-transparent compositing. These overlays revealed both well-aligned regions, characterized by crisp structural agreement, and areas of misalignment or ghosting, typically emerging in scenes with large viewpoint changes or insufficient inliers. In addition, an overlap mask was computed to explicitly mark regions where both images contributed valid pixel data, forming the basis for the blending stage.

To create the final panoramas, three blending strategies were implemented and compared: copy blending, average blending, and feather blending. Copy blending simply overwrites the reference image with warped pixels inside the overlap, often yielding sharp and visually distracting seams. Average blending smooths this transition by computing the pixelwise mean of the two sources, which reduces seam artifacts but introduces ghosting if the alignment is imperfect. Feather blending, in contrast, produces substantially more coherent results by computing a continuous distance-based weight map that gradually transitions contributions from each image toward the center of their respective regions. This soft, spatially varying weighting reduces brightness jumps, minimizes ghosting, and preserves structural sharpness, making it the most visually convincing method across the majority of scenes. While more challenging datasets such as v_graffiti still exhibited noticeable distortion due to parallax and homography failure, scenes like v_boat, v_circus, and v_weapons produced clean, well-aligned panoramas with smooth seams and consistent appearance.

I put some visual result from bird scene.



1_5_average blend

1_5_copy blend

1_5_feather blend

1_5_overlap_mask

1_5 overlay

1_5 panorama

Although multiple blending techniques were implemented to study their visual behavior, the panoramas presented in this report and used for further processing are exclusively based on feather blending. Copy blending and average blending were kept as diagnostic outputs to illustrate seam visibility and to demonstrate the effect of simple photometric fusion methods. However, copy blending produces abrupt transitions in the overlap region, and average blending often introduces ghosting when slight geometric misalignments remain. Feather blending, in contrast, computes a smooth spatial weighting using distance transforms, which gradually reduces the contribution of pixels near boundaries and increases the influence of central, more reliable regions. This results in significantly cleaner seams, smoother exposure transitions, and fewer perceptual artifacts. For this reason, the final panorama files saved as *_panorama.png correspond to the feather-blended result. All quantitative and qualitative evaluations in later sections—including the link to augmented reality—rely on these feather-blended panoramas, which consistently provided the highest fidelity across all scenes.

Overall, this stage demonstrates the effectiveness of combining robust homography estimation with carefully designed warping and blending strategies. The warped images, overlays, overlap masks, and final panoramas collectively illustrate the full geometric and

photometric pipeline required for high-quality image stitching. Moreover, this process establishes the essential foundation for the augmented reality task that follows: the same principles of per-frame homography estimation, geometric warping, and seamless compositing will be directly applied to dynamically project a video sequence onto a moving planar surface, ensuring that virtual content remains visually and geometrically coherent throughout the AR sequence.

**7.Augmented Reality (AR) via Per-Frame Homography Estimation:**

In the final stage of the assignment, i extended homography estimation pipeline to construct a dynamic Augmented Reality (AR) system. The goal of this component was to embed the frames of an external video sequence onto a planar object—specifically the cover of a book—visible in a moving target video. This task requires computing a new homography for each frame of the target video and warping the source video accordingly, ensuring that the embedded content remains rigidly attached to the book surface throughout the entire sequence.

To achieve this, İ first extracted SIFT keypoints and descriptors from the reference image (cv_cover.jpg) and from each incoming frame of book.mov. Feature matches were obtained via ratio-tested, cross-checked nearest-neighbor matching, producing highly reliable correspondences even under motion blur and viewpoint variation. These matches were passed to RANSAC-based homography estimator, which robustly rejected outliers and produced a per-frame projective mapping $H_t$that relates the cover image to the current video frame. The symmetric reprojection error was used as the inlier metric, which helped maintain geometric stability across frames.

Each frame of the source video (ar_source.mov) was center-cropped to match the aspect ratio of the book cover and then warped using $H_t$via cv2.warpPerspective. This operation transforms the synthetic content into the perspective of the book as seen in that frame. A corresponding binary mask was warped using the same homography to define the valid region of projection. The final AR frame was obtained by compositing the warped source image onto the book frame using masked overlay. This ensures that only the transformed video appears on the planar surface, while the rest of the scene remains untouched.

Representative results clearly demonstrate that the embedded video remains aligned with the book despite significant camera motion. Frames captured at timestamps t=0, t≈1s, t≈2s, and t≈4s show that the projected content correctly follows changes in scale, orientation, and perspective, indicating that the per-frame homography computation is successful. Minor jitter is observable in frames with limited detectable features due to motion blur; however, the overall sequence maintains geometric coherence and produces a visually convincing AR effect. The final output—a complete augmented video—exhibits smooth temporal behavior and demonstrates the practical application of homographies in real-time AR systems.

**8.Discussion: Accuracy, Robustness, and Limitations:** The overall pipeline—from feature detection to homography estimation, panorama stitching, and augmented reality compositing—works reliably when the underlying geometric assumptions are satisfied and the visual content contains sufficient texture for stable feature matching. In well-behaved scenes such as v_boat,

v_circus, and v_weapons, the planar surfaces are photographed under moderate viewpoint changes, and they contain rich, distinctive local features. In these cases, SIFT keypoints are detected densely and consistently, leading to high inlier ratios and low reprojection errors. The resulting homographies align the images accurately, producing panoramas with minimal seams and AR overlays that remain rigidly attached to the book surface across frames.

However, the pipeline begins to struggle when the planarity assumption is violated. In scenes like v_graffiti or in large baseline pairs (e.g., 1→6 in v_bird), significant parallax effects arise due to camera translation relative to 3D scene depth. Homography is only valid for planar scenes or pure rotational motion; when this assumption breaks, no single projective transformation can simultaneously align all image regions. This manifests as:inconsistent warping,ghosting or tearing in stitched panoramas,and, in extreme cases, RANSAC failures due to insufficient geometric consensus (e.g., *v_graffiti 1→6*).

Another source of difficulty is repetitive or low-texture regions, such as the graffiti pattern itself. These produce ambiguous descriptors and unreliable matches, which reduce inlier ratios and increase RANSAC iteration counts. Although SIFT is more robust than ORB, repetitive structures still trick the matcher into producing geometrically inconsistent correspondences.

Illumination differences and exposure changes between frames also degrade blending performance. Copy blending often produces hard seams, while average blending may introduce brightness shifts or double edges when alignment is imperfect. Feather blending mitigates these effects, but cannot compensate for geometrically invalid alignment caused by parallax or incorrect homographies.

In the AR stage, motion blur, rapid viewpoint changes, and frame-to-frame texture distortion create instability in homography estimation. Even when individual frames are processed correctly, small fluctuations in feature matches produce jittering overlays. The use of symmetric reprojection error, increased RANSAC iterations, and large inlier thresholds helps reduce this jitter, but some temporal flicker remains unavoidable without advanced tracking (e.g., optical flow or Kalman filtering). Nevertheless, for most frames, the AR projection remains stable and visually coherent, demonstrating that the core pipeline is sufficiently robust for real-time planar augmentation.

Overall, the system performs best when surfaces are planar, textures are distinctive, and the viewpoint changes smoothly. Its main limitations arise from fundamental geometric constraints of homography-based alignment and the variability of real-world video capture conditions. Despite these challenges, the implemented pipeline delivers accurate panoramas and convincing AR overlays in a wide range of scenarios.

**9.Reproducibility Notes**

To ensure that all experiments in the project can be reproduced reliably, the key parameters, random seeds, and implementation settings were fixed and documented throughout the pipeline. Although detailed parameter explanations were given in earlier sections, this subsection summarizes the most essential reproducibility criteria in one place.

Feature Extraction & Matching

Feature Detector: SIFT (default)

Descriptor Matching: k-NN with Lowe's Ratio Test

Ratio Test Threshold: 0.75

Cross-Check Matching: Enabled

ORB variant: Used only experimentally for AR timing comparisons

Keypoint normalization: Disabled for SIFT (built-in scale invariance), applied in DLT stage

Homography Estimation (DLT + RANSAC)

Point Normalization: Hartley normalization (meanzero + scale √2)

Minimal Sample Size: 4 correspondences

RANSAC Maximum Iterations: 2000

Inlier Threshold: 3.0 pixels (symmetric reprojection error)

Confidence Level: 0.995

Random Seed: 1337 (fixed for every run)

Refinement Step: Recompute homography using all inliers after RANSAC

Panorama Construction

Warping: cv2.warpPerspective with bilinear interpolation

Canvas Padding: Automatic via warped corner bounding box

Blending Modes Implemented:

Copy blending

Average blending

Feather blending *(distance-transform weights)*

Final Selected Panorama for Report: Feather-blended output

Augmented Reality (AR) Pipeline

Frame Step: 1 (every frame processed; optionally 2 for faster runtime)

Book Video FPS: Read dynamically from video metadata

Source-to-Cover Cropping: Center-crop + resize to cover aspect ratio

Homography Per Frame: SIFT → Ratio Test → RANSAC

Compositing: Mask-based replacement using warped source frame

Output Video Codec: mp4v

Output Frame Size: Same as book.mov resolution

Link for ar_dynamic_result.mp4:
https://drive.google.com/file/d/1_AHXezN2Cl_jGF_NxxC21tUSaEHjCoVq/view?usp=drive_link

Link for demo:
https://drive.google.com/file/d/1pdSVIVkK65h0Ti9SbHzgVHXdRe3WUJkg/view?usp=sharing

Panorama_results link: https://drive.google.com/drive/folders/1-_9nfAVdcCmy-nuykwCeACYG0POzHsfj?usp=sharing

Link for .py files : submit system does not accept .py files so i give drive link for this .py codes:
https://drive.google.com/drive/folders/1XC7gt-_FhOBvAKb79SAhk2obAbhfWELA?usp=sharing

Notes

In jupyter notebook i do not show output images results i show them in outputs folder. also i used .py files with juypter notebook i locate.py files on folder path modules/*.py files i locate data files also in same place. My local path: C:\Users\mbdn1\Desktop\BBM418_Assignment2 so i run jupyter notebook inside the BBM418_Assignment2.