CPS843 HW4

Saihaan Syed 501105781
saihana.syed@torontomu.com

***Abstract*—My solutions to CPS843 homework 4**

Part 1

Problem 1 — Plane at infinity and plane transforms

1. **Canonical plane at infinity.** In homogeneous 4D coordinates a plane is represented by a vector $\pi = (\mathbf{n}^\top, \rho)^\top$ such that $\pi^\top X = 0$. The plane at infinity contains every direction $X_\infty = (\mathbf{d}^\top, 0)^\top$, so substituting $\rho = 0$ and $\mathbf{n} = \mathbf{0}$ yields the canonical form $\pi_\infty = (0, 0, 0, 1)^\top$. Any point with last coordinate zero satisfies $\pi_\infty^\top X = 0$, so this vector captures "all points at infinity."
2. **Affine transforms fix π_∞.** A 3D affine transform has homogeneous matrix $H = \begin{bmatrix} A & \mathbf{t} \\ 0^\top & 1 \end{bmatrix}$ with invertible $A \in \mathbb{R}^{3\times 3}$. Applying H^{-T} to π_∞ gives
$$H^{-T}\pi_\infty = \begin{bmatrix} A^{-T} & 0 \\ -\mathbf{t}^\top A^{-T} & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \pi_\infty.$$
Therefore the plane at infinity is invariant under every affine transformation: directions stay directions, so the distinction between finite and infinite points is preserved.
3. **Plane transformation rule.** A 3D point transforms as $X' = HX$. For any plane π we require $\pi'^\top X' = 0$ whenever $\pi^\top X = 0$. Substituting $X' = HX$ gives $\pi'^\top HX = 0$ for all X on the original plane, which is satisfied when $H^\top \pi' = \pi$ or equivalently $\pi' = H^{-T}\pi$. Thus planes transform by the inverse-transpose of the point transform, matching the projective duality between points and planes.

Problem 2 — Homographies implied by special motions

1. **Coplanar scene homography.** If all 3D points lie on plane Π with equation $\mathbf{n}^\top X + d = 0$, choose plane coordinates $X = X_0 + \alpha r_1 + \beta r_2$ so that $[r_1\ r_2\ X_0]$ spans Π. Their projections through camera $P = K[R \mid t]$ become
$$x \sim K(R[r_1\ r_2\ t] \begin{bmatrix} \alpha \\ \beta \\ 1 \end{bmatrix}),$$
which shows image points are related by the planar 2D homography $H = K[r_1\ r_2\ t]$. Hence any two images of the same plane differ by a single 3×3 matrix.
2. **Back-projecting an image line.** An image line l satisfies $l^\top x = 0$ for all projections $x = PX$. Substituting x yields $l^\top PX = 0$, or $(P^\top l)^\top X = 0$. Therefore the 4-vector $\Pi = P^\top l$ defines a 3D plane that contains every 3D point mapping to that image line—precisely the back-projection plane through the optical center.
3. **Zoom-only motion.** A pure zoom changes only the focal length (and possibly principal point due to digital rescaling) but not the extrinsics, so $P = K[R \mid t]$ and $P' = K'[R \mid t]$. Eliminating the common rigid term yields
$$x' \sim P'X \sim K'[R \mid t]X \sim K'K^{-1}(K[R \mid t]X) \sim Hx,$$
with $H = K'K^{-1}$. Thus consecutive zoomed images are homographically related by the product of the new and old intrinsic matrices.

Problem 3 — Image of the absolute conic (IAC)

1. **Expression of ω.** The absolute conic Ω_∞ lives on π_∞ and is mapped to the image by $PK = K[I \mid 0]$. Applying the projection yields $\omega = K^{-\top}K^{-1} = (KK^\top)^{-1}$, a symmetric positive-definite matrix encoding intrinsic parameters only.
2. **Orthogonality via ω.** A world pair of orthogonal directions X_1, X_2 satisfies $X_1^\top X_2 = 0$ in Euclidean coordinates. Their projections are $x_i \sim KRX_i$, so $x_1^\top (KK^\top)^{-1} x_2 = X_1^\top R^\top R X_2 = X_1^\top X_2 = 0$. Hence $x_1^\top \omega x_2 = 0$ characterises orthogonal image rays.
3. **Constraints from a plane-to-image homography.** For a plane with homography $H = [h_1, h_2, h_3]$, the columns h_1 and h_2 correspond to projections of two orthogonal directions on the plane, while h_3 comes from the plane's origin. Because columns inherit the Euclidean structure of the plane, we obtain two constraints:
$$h_1^\top \omega h_2 = 0, \quad h_1^\top \omega h_1 = h_2^\top \omega h_2.$$
Each supplies a linear equation in the unknown entries of ω.
4. **Square-pixel assumption.** Square pixels imply zero skew and equal focal lengths, so
$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \omega = \left[\begin{matrix} 1/f^2 & 0 & -c_x/f^2 \\ 0 & 1/f^2 & -c_y/f^2 \\ -c_x/f^2 & -c_y/f^2 & (c_x^2 + c_y^2 + f^2)/f \end{matrix}\right.$$
The zeros in the off-diagonal entries impose the familiar constraints $\omega_{12} = 0$ and $\omega_{11} = \omega_{22}$, which

directly encode the square-pixel prior used in self-calibration.

Part 2 — Apply SIFT

SIFT Technical Overview

I went through the Lowe "Distinctive Image Features" paper and the Brown & Lowe panorama paper. SIFT (Scale-Invariant Feature Transform) finds little landmark patches in a photo that stay recognisable even if another picture is taken closer, further away, or with the camera tilted.

- **Step 1 – build a scale space.** The algorithm repeatedly blurs the image with Gaussians and subtracts neighbouring blur levels (Difference of Gaussians). By scanning that 3D scale-space volume for strong peaks and valleys compared to the 26 surrounding samples, we collect candidate keypoints at different zoom levels.
- **Step 2 – keep only stable extrema.** Tiny contrast blobs or points that just sit on straight edges are thrown out. This happens by checking the DoG strength and the Hessian matrix, just like the papers describe, so the remaining keypoints are reliable under real-world noise.
- **Step 3 – assign each keypoint an orientation.** Around every surviving point we measure gradient directions and vote in an orientation histogram. The biggest peak becomes the keypoint's reference angle, which means the descriptor can be rotated to match cameras that were held sideways or upside down.
- **Step 4 – build the 128-value descriptor.** The neighbourhood is rotated and split into a 4×4 grid. Within each cell we collect an 8-bin gradient histogram, then normalise and clamp the combined 128-D vector. Using gradients plus normalisation is what makes the descriptor shrug off lighting changes and flash shots, something the panorama paper relies on for fully automatic matching.

For **matching**, the panorama pipeline drops all descriptors into a k-d tree, looks up the nearest and second-nearest neighbours, and only keeps a match when the ratio of distances is small (the closest match is clearly better). Brown & Lowe then run RANSAC on those tentative matches to estimate a homography and filter out remaining outliers before bundle adjustment refines all camera parameters. Because SIFT is already invariant to rotation, scale, and moderate lighting changes, the software can stitch unordered handheld images without any manual alignment.

TLDR: "find strong scale-space blobs, give them a stable angle, summarize the rotated gradients into a normalised vector, and compare those vectors smartly so panorama tools can line up overlapping photos."

Practical Application

I used the Autostitch application from https://mattabrown.github.io/autostitch.html as opposed to the link in the assignment file as the download link there is broken.

Input Images

Output

Brief Analysis and Discussion of Results

Looking at the six input frames, the software always has plenty of shared features to grab onto. The lighting is mostly consistent because the pictures seem as though they were taken during the same outing, which helps SIFT keep the gradient patterns stable.

The stitched panorama looks like a single wide shot to my eye. Straight objects such as the building edge stay pretty straight. The only small issue I notice is it struggling with the exit light probably due to the different blur around there for all the images.

Part 3 - Group Project Progress

We have divided the research papers and are individually running and evaluating the different algorithms.