CS 83/183: Assignment 2

February 4, 2024

Jessie Li

Question 2.1

1. **h** has 8 degrees of freedom.

2. 4 point pairs are required to solve **h**.

3. $$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

   Multiply:

   $$\begin{aligned} x_1 &= \alpha(h_1 x_2 + h_2 y_2 + h_3) \\ y_1 &= \alpha(h_4 x_2 + h_5 y_2 + h_6) \\ 1 &= \alpha(h_7 x_2 + h_8 y_2 + h_9) \end{aligned}$$

   Divide the first two rows by the last to cancel α:

   $$\begin{aligned} x_1 \cdot (h_7 x_2 + h_8 y_2 + h_9) &= (h_1 x_2 + h_2 y_2 + h_3) \\ y_1 \cdot (h_7 x_2 + h_8 y_2 + h_9) &= (h_4 x_2 + h_5 y_2 + h_6) \end{aligned}$$

   Move right hand side to the left:

   $$\begin{aligned} h_7 x_2 x_1 + h_8 y_2 x_1 + h_9 x_1 - h_1 x_2 - h_2 y_2 - h_3 &= 0 \\ h_7 x_2 y_1 + h_8 y_2 y_1 + h_9 y_1 - h_4 x_2 - h_5 y_2 - h_6 &= 0 \end{aligned}$$

   Expressed as $\mathbf{A_i h} = 0$:

   $$\mathbf{A_i} = \begin{bmatrix} -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2 x_1 & y_2 x_1 & x_1 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2 y_1 & y_2 y_1 & y_1 \end{bmatrix}$$

   for each point pair i (2 equations per correspondence).

4. The trivial solution for **h** would be the zero vector. If we have m points, **A** would be an $2m$ x 9 matrix and rank(**A**) $\leq \min(2m, 9)$.

   - If rank(**A**) $= 8$, **A** would have full row rank but not full column rank. 0 would be a singular value of **A** and the corresponding singular vector (a vector in the 1D null space of **A**) would be a unique solution for **h** up to scale.
   - If rank(**A**) < 8, the system would be under-determined and there would be infinitely many solutions for **h**.
   - If rank(**A**) $= 9 > 8$, the only solution to $\mathbf{Ah} = 0$ would be the trivial solution. We would use SVD or some other method to find a solution **h** that minimizes $||\mathbf{Ah}||$ subject to $||\mathbf{h}|| = 1$.

   With 4 point pairs, rank(**A**) ≤ 8. In most cases, we can assume rank(**A**) $= 8$ and a unique h exists because the four point pairs are unlikely to be collinear. If we have more than 4 pairs of points, the third case could apply.

Question 3.1: FAST Detector

Reference:

- FAST Algorithm for Corner Detection (OpenCV)

The Harris corner detector finds corners by looking for pixels where *changes* in intensity are large in every direction. For each pixel, the Harris corner detector calculates the x and y image gradients for all pixels over a small region, subtracts the mean, then computes the gradient covariance matrix $\mathbf{M}$. $\mathbf{M}$ fits a quadratic to the shape of the gradients, and its eigenvalues and eigenvalues tell us the rate and direction of fastest and slowest change. Corners are detected by applying a threshold to the eigenvalues (like Harris and Stephens'). Corners generally resemble bowl-shaped paraboloids with both eigenvalues similar in magnitude and large, indicating that the intensity changes quickly in all directions.

FAST looks at intensities around a center pixel, rather than changes in pixel intensity. If n contiguous pixels around a center pixel p are all brighter or darker than p by some threshold t, then p is a corner.

FAST is computationally more efficient because it performs far fewer calculations per pixel. But Harris, which computes the gradients, covariance matrix, and eigenvalues for each pixel, is likely more robust to noise.

Question 3.2: BRIEF Descriptor

References:

- BRIEF: Binary Robust Independent Elementary Features (Calonder et al.)
- BRIEF (Binary Robust Independent Elementary Features) (OpenCV)

BRIEF provides fast and memory-efficient feature matching. After feature detection (FAST in our case), BRIEF converts image patches around interest points into binary descriptors based on pixel intensity comparisons. Other methods that calculate the full descriptor, such as SIFT (128-dimension descriptor vectors) and SURF (64-dimension descriptor vectors), consume more time and memory, but may be more robust to transformations.

BRIEF first creates and smooths image patches around the given interest points to reduce noise sensitivity. For each image patch, it chooses a unique set of n pixel pairs (x, y) and creates an n by n binary feature vector τ such that $\tau(x, y) = 1$ where the intensity at x is less than the intensity at y, 0 otherwise. n can be 128, 256 or 512. Once this is done, BRIEF uses Hamming distance to match descriptors.

Whereas BRIEF creates binary descriptors by comparing pixel intensities, other methods we saw in class use filter banks to build their descriptors. Examples include the Haar wavelets (used in MOPS) and Gabor filters (used in GIST). These filters may be better at capturing certain types of features in images.

Question 3.3: Matching Methods

References:

- BRIEF: Binary Robust Independent Elementary Features (Calonder et al.)

- Hamming Distance (Danziger)

The Hamming distance between two binary vectors is the number of bits that differ. Hamming distance can easily be computed by performing an element-wise XOR of the two vectors then counting the nonzero bits.

Given two images, BRIEF picks n interest points from both images and calculates their binary descriptors. Each point a in the first image is matched with the nearest neighbor b in the second image that minimizes the Hamming distance between a and b.

Hamming distance makes sense for binary BRIEF descriptors and is computationally more efficient than Euclidean distance for real-valued vectors.

Question 3.4

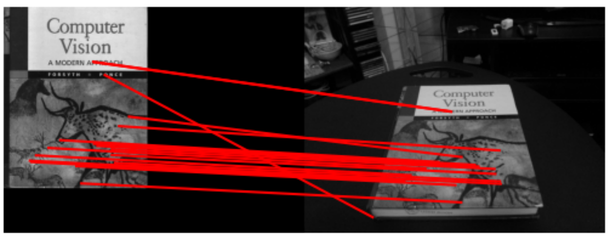

Figure 1: Points matched in `cv_cover.jpg` and `cv_desk.png`.

Question 3.5

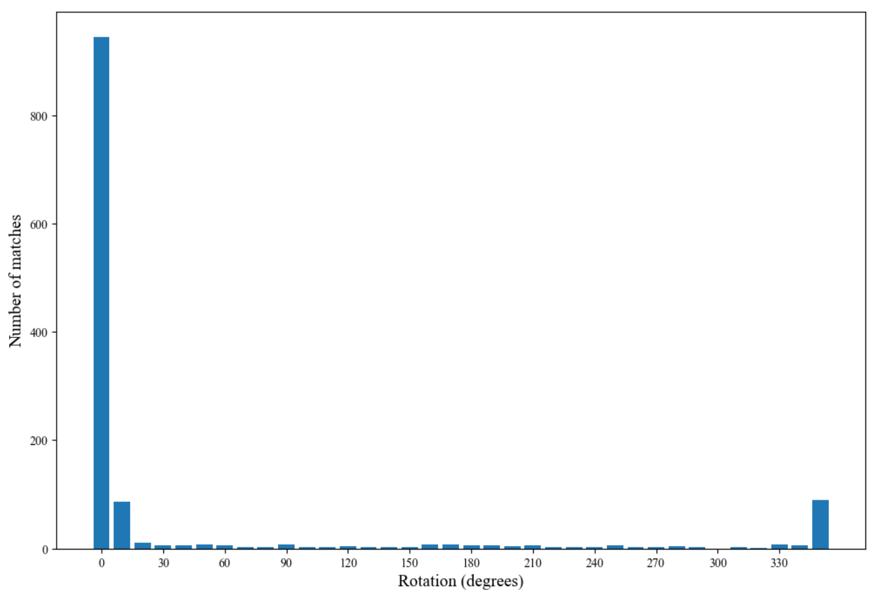

Figure 2: Count of matches for each orientation (`sigma` = 0.15 and `ratio` = 0.7)

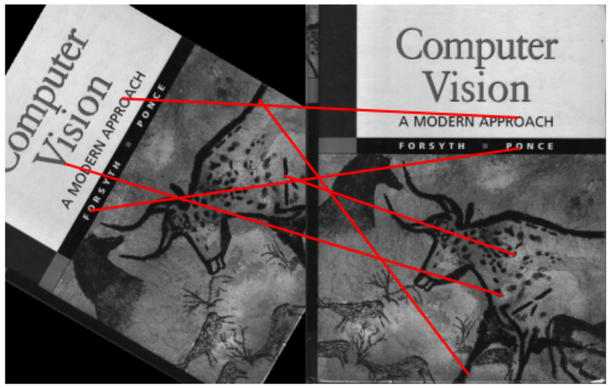

Figure 3: 60 degree rotation (5 matches).

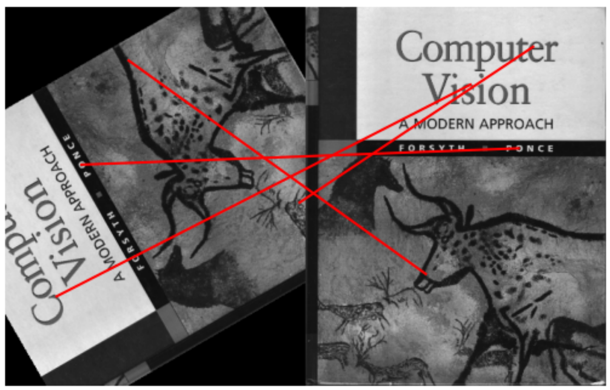

Figure 4: 120 degree rotation (4 matches).

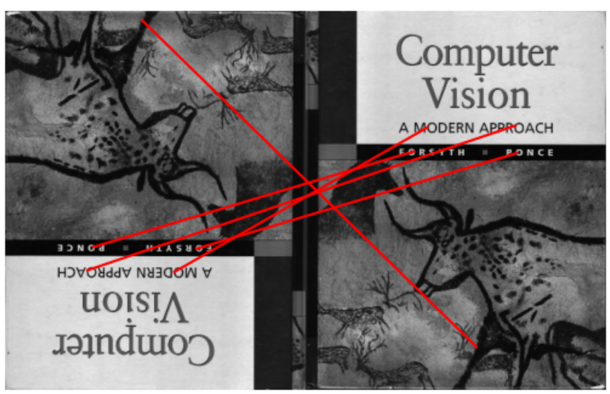

Figure 5: 180 degree rotation (5 matches).

With no rotation, BRIEF found 945 matches. As seen in Figures 3, 4, and 5, there were considerably fewer matches for all other orientations, showing that BRIEF is not rotationally invariant. This makes sense because BRIEF compares pixel pairs within a small patch around an interest point, and these pairs are fixed regardless of how the image is oriented. Rotating the image rotates the patches, leading to different binary strings and changing the Hamming distance between each pair of points.

Question 3.8: RANSAC

- `d` = 1 (distance threshold)
- `iter` = 200 (number of samples)
- `s` = 4 (number of points per sample)

Question 3.9: Harry-Potterized Textbook

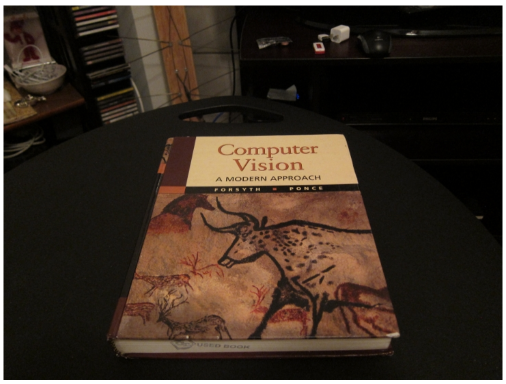

Figure 6: Original textbook

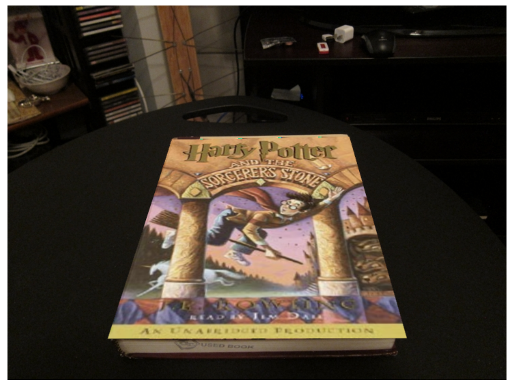

Figure 7: Composite image

`hp_cover` must be resized to match the dimensions of `cv_cover` before applying the homography. The homography uniquely maps coordinates from `cv_cover` to `cv_desk`, so if the dimensions of `hp_cover` and `cv_cover` do not match, the shape of the warped images would also be different.

Extra Credit

`ar_ec.py`

See `ec/ar.avi`.

Figure 8: Sample from the output video.

`panaroma.py`

Figure 9: Left image

Figure 10: Right image

Figure 11: Panaroma