CS7.505 - Mid-Semester Exam

Avneesh Mishra
avneesh.mishra@research.iiit.ac.in *

March 5, 2022 to March 6, 2022

Contents

*M.S. by research - CSE, IIIT Hyderabad, Roll No: 2021701032

1 Questions

I formulated the following questions for the exam.

1. Enunciate any two methods of keypoint detection and description. Can these be used in an ensemble? If yes, then how?
2. Describe and derive the Fundamental Matrix, Essential Matrix, and Homography Matrix (for the case of pure rotation). When there are many correspondences between two images, can these methods be used to filter out the best correspondences?

1.1 Reasons

The reasons why I feel these questions are worthy and/or interesting

Q1: Keypoint detection and description

Question

> Enunciate any two methods of keypoint detection and description. Can these be used in an ensemble? If yes, then how?

Reason This question promotes a deeper dive into the reading material for the theoretical methods taught in class. It should serve as a quick and good reference for traditional keypoint detection and description methods. The aim is to have a good information archive, created through reading the original text, well summarized, in one place.

Ensemble techniques have recently caught steam, especially in the age of deep learning. Exploring such options for traditional methods could yield stronger baselines for traditional feature detection and description methods.

Q2: Relating matrices and RANSAC

Question

> Describe and derive the Fundamental Matrix, Essential Matrix, and Homography Matrix (for the case of pure rotation). When there are many correspondences between two images, can these methods be used to filter out the best correspondences?

Reason This question is to lay a foundation for matrices relating to two images. Though the derivation is not traditionally important, it is good to have the theoretical backbone in one place. Random Sample Consensus (RANSAC) is a very popular method to boost the performance of a correspondence matching algorithm. This question aims to derive the theory behind it and also give a direction on how it can be applied using the knowledge of these basic matrices in computer vision. It concludes with examples and references to a python package that can perform RANSAC using the matrices.

2 Q1: Keypoint Detection and Description

Question

> Enunciate any two methods of keypoint detection and description. Can these be used in an ensemble? If yes, then how?

Keypoints are points of *interest* and are useful in image matching and description. Keypoints have to be *detected* (location found in an image) by a detector, and they have to be described by a *descriptor* (for some unique identification).

The answer is described in subsections below.

2.1 SIFT

Scale Invariant Feature Transform (SIFT) is an image feature generation method introduced by David G. Lowe in [Low99]. A more explained iteration was presented in [Low04] with some revisions to the model. The primary contribution of the author was exploring the features in multiple scales (through an image pyramid), which makes the keypoint descriptors *scale* invariant.

Detector The keypoint detector has the following basic steps

1. Construction of DoG (Difference of Gaussian) image pyramid: The input image resolution is increased (scaled up) by a factor of two using bilinear interpolation. Then two successive gaussian blurs are applied, yielding image `A` (less blurred) and `B` (more blurred). Subtracting image `B` from `A` given the Difference of Gaussian image. This is repeated for scales of 1.5 in each direction (up and down). This scaling factor was later revised to 2.

2. Achieve keypoint locations: The local extrema in the DoG image is a keypoint. First, eight neighbor comparisons are made at the same scale. Then, if the point is in extrema (maximum or minimum), comparisons are made at higher scales (position interpolation to maintain scale).

3. Extract Keypoint orientations: The image `A` is used to compute the gradient magnitudes and orientations. The magnitudes are thresholded to 0.1 times the maximum gradient value (to reduce illumination effects). A histogram of gradient orientations in the local neighborhood of keypoints is created. The weight of the orientations is the thresholded gradient values. This histogram (containing 36 bins covering 0° to 360°) is smoothened, and the peak is chosen as the gradient orientation.

In the end, the keypoint locations (on the image) and orientations are obtained. The direction is used to achieve rotation-invariant features.

Descriptor The keypoint descriptor (as presented in the original work in [Low99]) has the following basic steps

1. Reorient the local region around the keypoint. This is basically to set the local orientation (of keypoint) as a reference. This is done by simple subtraction of gradient orientations in later steps.

2. Subdivide the local region: The local region (within the radius of 8 pixels) is sub-divided into a 4×4 sub-array, with each sub-array having an 8-bin gradient histogram.

3. Run the same on a larger scale version: On one scale higher in the pyramid, perform the above step but with a 2×2 sub-array (still 8 bins in the histogram).

Note that the gradient directions for the histogram are not just the gradients at the center pixel but are interpolated in the $n \times n$ grid (sub-array with $n = 4$ or 2). The total number of SIFT descriptors (length) for a keypoint is $8 \times 4 \times 4 + 8 \times 2 \times 2 = 160$.

In the revised edition [Low04], a 16×16 local region is sub-divided into 4×4 grid, with each grid having 8 orientation bins (from histogram). Therefore, the new descriptor length becomes $8 \times 4 \times 4 = 128$. It is found that this is much faster to compute and doesn't have a large compromise on performance.

2.2 SURF

Speeded-Up Robust Features (SURF) is another feature detection and description algorithm proposed by Herbert Bay, et al. in [BTG06]. This was also described in a more illustrated manner in [Bay+08]. The primary contribution of the authors were exploiting the idea of integral image (which Voila and Jones originally proposed in [VJ01]), to speed up calculation of an approximated second order Gaussian (the Hessian matrix). The authors also proposed robust methods for descriptor extraction.

Detector The keypoint detector has the following basic steps

1. Estimate the integral image for the input image, using the equation below.

$$I_{\sum}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i,j)$$

2. Approximate the terms in the hessian matrix: The determinant of hessian matrix is used as a proxy for feature points. Instead of using the Difference of Gaussian to estimate the terms in matrix, a second order approximation with simple terms is used.

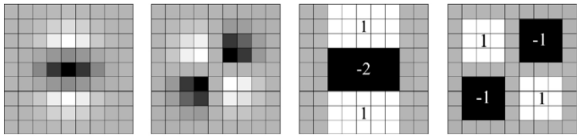

Figure 1: Approximation of second order derivatives
Left to right: Instead of L_{yy} and L_{xy} (first two), we use D_{yy} and D_{xy}.

This approximation requires yields feature value as $\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2$. This is much faster to compute, but is still single scale

3. Multiple scales: Instead of resizing the image, we can resize these kernels (as shown in the figure above). The standard sizes (to preserve center pixel) are 9×9, 15×15, 27×27, ... An octave consists of a series of filter response maps obtained using convolution with different filter sizes (set of four usually successive sizes).

4. Non-Maximum Suppression: The maxima values in a $3 \times 3 \times 3$ neighborhood are retained and these are interpolated to their true scale and position on image.

This yields the position of the keypoints; not just on the image, but also the particular scale of detection (s).

Orientation alignment We need to obtain orientations before getting the descriptors. This is done in the following steps

1. Calculate the *Haar-wavelets* in the local region around the keypoint: We approximate the dx and dy filter (gradient in X and Y direction respectively) with a kernel containing $+1$ and -1 values. This is run on the $6s \times 6s$ neighborhood of the keypoint to get the gradients of neighboring points.

2. Apply a gaussian weight with $\sigma = 2s$

3. Represent each point in the neighborhood as a point in a 2D scatter plot with X and Y values being the weighed dx and dy values.

4. On this 2D scatter plot, run a window in polar form with the angle being $\pi/3$. Get the window with maximum sum (of weights of the points in the window).

5. For this window, the orientation is calculated by summing the X and the Y values of the points (separately) and then getting the angle.

We now have the orientation of each keypoint (thereby allowing us to get rotation robust descriptors). This orientation is also linked to the same scaling factor in which the keypoint was detected.

Descriptor The descriptor is calculated in the following steps

1. Get a $20s \times 20s$ oriented square patch around the keypoint (centered at the local feature). Calculate the integral image for this oriented patch, and estimate the Haar-wavelets (similar to the orientation alignment part) for dx and dy values for each pixel in this patch.
2. Split this patch into 4×4 sub-regions, with each sub-region having 5×5 samples (actually, $5s \times 5s$ pixels).
3. For each sub-region, calculate $\mathbf{v} = [\sum d_x, \sum |d_x|, \sum d_y, \sum |d_y|]$, a 4-dimensional descriptor of the particular sub-region.
4. Stacking these 4-dimensional descriptors for every sub-region into a column vector gives the SURF descriptor. Invariance to contrast is achieved through normalizing them.

The traditional SURF algorithm therefore gives a descriptor of length $4 \times 4 \times 4 = 64$.

Despite being of smaller length, the descriptor (along with the matching method described in section 4.3 of [Bay+08]) seems to give more robust correspondences than most other then-state-of-the-art methods. The authors demonstrate 3D reconstruction from un-calibrated cameras in section 5.2 of [Bay+08].

2.3 Ensemble

TL; DR It depends on the application. Let us take the application of *finding feature correspondences between two images* as an example.

Example The aim is to match the identical features in two images. Feature description plays an essential role here.

Traditionally, the descriptors are uniquely defined for each method (SIFT and SURF, for Example, have different descriptor formats). They, therefore, cannot be concatenated or merged in any easy way.

However, we can apply some tricks to get an ensemble of correspondences. One of them is to apply descriptor matching (using techniques like mutual nearest neighbor, cosine distance, Euclidean distance, or Mahalanobis distance) for the *individual* methods (separately). Then, obtain the keypoints (again, separately) and then concatenate the obtained keypoints. We now have point correspondences from both methods.

Such methods can boost correspondences between two images by a significant margin.

References

[Low99] David G Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[VJ01] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. Ieee. 2001, pp. I–I.

[Low04] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[BTG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features". In: *European conference on computer vision*. Springer. 2006, pp. 404–417.

[Bay+08] Herbert Bay et al. "Speeded-up robust features (SURF)". In: *Computer vision and image understanding* 110.3 (2008), pp. 346–359.