# Feature Detectors and Descriptors and Motion Estimation

Authors: Manimozhi Ramasamy, Pragna Munagala

Group - 7

## Introduction:

In computer vision and image processing the concept of feature detection refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. Types of image features include Edges, Corners, Blobes and Ridges. Once features have been detected, a local image patch around the feature can be extracted. This extraction may involve quite considerable amounts of image processing. The result is known as a feature descriptor or feature vector.

Transformation from one 2D image to another is described using motion vectors, in videos it is described using adjacent frames. This process is known as Motion Estimation in image processing. Motion vectors may related to whole image or specific parts of the image. It depends on the type of feature keypoints and the descriptors we use for the estimation. Several different methods for motion estimation based on optical flow have been developed at Fraunhofer IOSB. Some methods allow real-time processing of large images in real-time on GPUs, whereas other methods allow for computation of more accurate and reliable estimates.

**Report Organization:**

- The Report is organized into four sections which include Adopted Feature Detectors and Descriptors including Motion Estimation, Results, Conclusion and References.
- In the next section, SIFT and SURF feature detectors are discussed including the KLT used for optical flow in Motion Estimation.

- In Results section, elapsed time of 1. SIFT and SURF feature detectors along with optical flow and 2. SIFT and SURF with Flann matcher are indicated. Also a sample of output screenshots are presented for an object. A brief comparison of output of both approaches is mentioned.
- In Conclusion Section, unsolved issues and future directions for possible improvements are discussed.
- In References Section, references used for implementation of feature detectors and descriptors and motion estimation are mentioned.

**Adopted Feature Detectors and Descriptors:**

**Importance of descriptors:** After extracting the keypoints, information about their position, and their coverage area in the image is only obtained. Depending on the algorithm used to extract keypoint (SIFT, SURF in our case), some general characteristics of the extracted keypoints is known but no information is known about how different or similar one keypoint is to the other. So, here come descriptors: they are the way to compare the keypoints. Some characteristics about the keypoints are specified by the descriptors in vector format of constant length. For example, it could be their intensity in the direction of their most pronounced orientation. The descriptors should be independent of keypoints position, robust against image transformations, scale independent. The keypoints whose descriptors have the smallest distance between them are matches, e.g. same "places" or "parts of objects" in different images.

**1. SIFT**

The detectors such as Harris are rotation invariant ( as the corners of the image are not affected by rotation). But when the image is scaled the corners in the original image may not the detected as corners in the scaled image. Hence SIFT detector is considered in this project.

SIFT - Scale Invariant Feature Detector converts the image data into scale invariant co-ordinate points relative to the local features. The SIFT detector is involved with 4 steps namely,

**Scale Space Extreme Detection**: For detecting large corners, the SIFT algorithm uses Difference of Gaussian. The Difference of Gaussian is obtained by Gaussian blurring of images with different scale parameter $\sigma$ and finding the difference between them. This process is repeated for every octave in image in the Gaussian pyramid. Any pixel in one octave layer is compared to its 8 nearby neighbors and nine other pixels in the above and below octave layer. If that pixel is the local extrema, then it is a potential keypoint.

**Keypoint Localization:** The potential keypoints are found from the previous steps. To obtain the accurate keypoint, Taylor series expansion is performed on the the scale space to find the local extrema. If the extrema value is less than the contrastThreshold ( 0.03 from [1]) , then that extrema is discarded. Since the edges give good response for Difference of Gaussian, they need to removed. A Hessian matrix of 2X2 is used to calculate the principal curvature. When the ratio of the Eigenvalues is greater than the edgeThreshold( defined as 10 from [1]), those keypoints are rejected.

**Orientation Assignment**: If a consistent orientation is assigned to every keypoint based on the local image properties, then the descriptors can be described relative to the local points. From the gradient orientation of the points near the key point, orient histogram is obtained and peak of the histogram corresponds to the dominant direction of local gradient.

**Keypoint Descriptors**: To obtain the keypoint descriptor, a neighbor of 16X16 is taken around the keypoint. The 16X16 neighbors are subdivided into 4X4 size. The 8-bin orientation histogram is performed on each subblock resulting in 128 bins. This defines the keypoint descriptors.

**Keypoint Matching**: The keypoints are matched between images by their nearby neighbors. When the noise in the image is high, the first and the second close match may be very near. Then the ratio of first close match distance to the second close match distance is taken, and if this ratio is greater than 0.8, they are rejected.

**OpenCV Command for SIFT:**

**Ptr<SIFT> detector = SIFT::create()** is used to create the sift detector.

**detector->detect()** is used to detect the keypoints in the image.

**detector->compute()** is used to compute the descriptors from the keypoints.

**2. SURF**

Speeded up robust features (SURF) is used for object recognition, image registration, classification or 3D reconstruction. The standard version of SURF is several times faster and more robust than SIFT against different image transformations. For the detection of interest points, SURF uses an integer approximation of the determinant of Hessian blob detector. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest. The good performance of SIFT compared to other descriptors is remarkable. Its mixing of crudely localised information and the distribution of gradient related features seems to yield good distinctive power while fending off the effects of localisation errors in terms of scale or space. The SURF descriptor is based on similar properties, with a complexity stripped down. The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. Then, a square region aligned to the selected orientation is constructed, and the SURF descriptor is extracted. One important feature of SURF is usage of Sign of Laplacian, which was computed during the detection. The Sign of Laplacian is used to identify dark blob on the bright background and vice versa. This improves the matching process without additional increase in the complexity. The SURF is almost 3 times faster than SIFT and has comparable performance as that of SIFT.

**OpenCV Command for SURF:**

**Ptr<SURF> detector = SURF::create()** is used to create the SURF detector.

**detector->detect()** is used to detect the keypoints in the image.

**detector->compute()** is used to compute the descriptors from the keypoints.

**3. Motion Estimation**

Motion estimation is an important part of image analysis. Differential methods based on the so called optical flow belong to the most accurate methods for motion estimation. Motion Field (projection of 3D motion onto 2D image plane) and Optical Flow (the apparent motion of the brightness pattern in image sequence) are the key terms in motion estimation.

Optical flow methods are based on the assumption that pixel-values between images change only because of motion. The main drawback of optical flow methods is the complexity of the algorithms which may lead to long processing times.

Lucas–Kanade(KLT) method is the widely used differential method for optical flow estimation. It assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion. KLT optical flow is less sensitive to noise than other pointwise methods, whereas as it doesn't provide flow information about interior uniform regions of the image as it is a purely local method. The motion estimation has many applications. The process of applying motion vector to obtain the transformation to next image is called motion compensation. Motion estimation and compensation forms integral element is video compression. Motion estimation are widely used in automatic speech recognition techniques for capturing lip movement.

**Motion Estimation Using Optical Flow Procedure:**

- The first frame of the video is loaded.
- SURF/SIFT detector is used to detect the keypoints in the first frame.
- Until end of the video is reached, following steps are executed:
    - For the first frame, the previous frame is empty. Hence first frame data is copied into previous frame object and remaining steps are skipped.

    - For the second frame, current frame is read and previous frame exist. calcOpticalFlowPyrLK is used to calculate the location of the keypoints in the current frame.
    - A line is drawn between the corresponding keypoints found in the previous frame to the current frame.
    - The frame is displayed along with the tracked lines.
    - Output is written into .avi file.

**Motion Estimation Using Feature Detectors and FLANN Matching Procedure:**

- Input video is loaded.
- SURF/SIFT detector is used for keypoint detection and descriptors in each frame.
- For the first frame, previous and current frame, descriptors and keypoints will be the same.
- For the remaining frames until the end of video is reached, following procedure is followed:
    - Keypoints and descriptors for the current frame are computed.
    - Flann matcher is used to find the matches between the current and previous descriptors.
    - Good Matches are calculated by finding the minimum and maximum distance and setting up a threshold.
    - Lines are drawn connecting the good matched in the previous frame to those in current frame.
    - Each frame is added onto the canvas output.
    - Previous descriptors, frame, keypoints are assigned the current descriptors, frame, keypoints values.
    - Current frame is displayed with the flow of lines.
    - Output is return into .avi file.

**Results:**

Complete results (videos) can be found in the output folder in the submitted zipped folder.

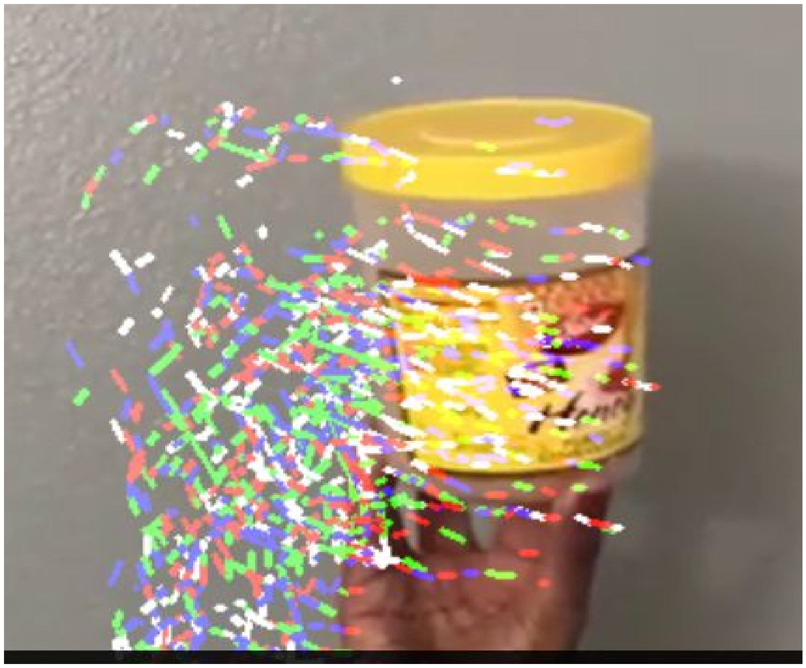
The object is moved initially in upwards direction and then towards right.

**SURF Keypoint :**

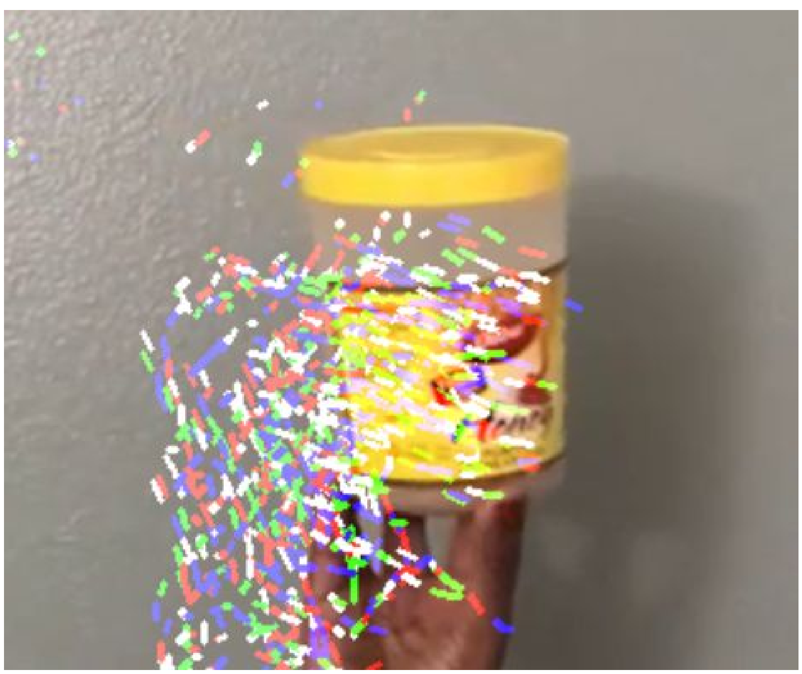
**SIFT Keypoints:**

**1. A part of results of using feature descriptors with Flann Matcher are as shown below,**

Using **SURF**

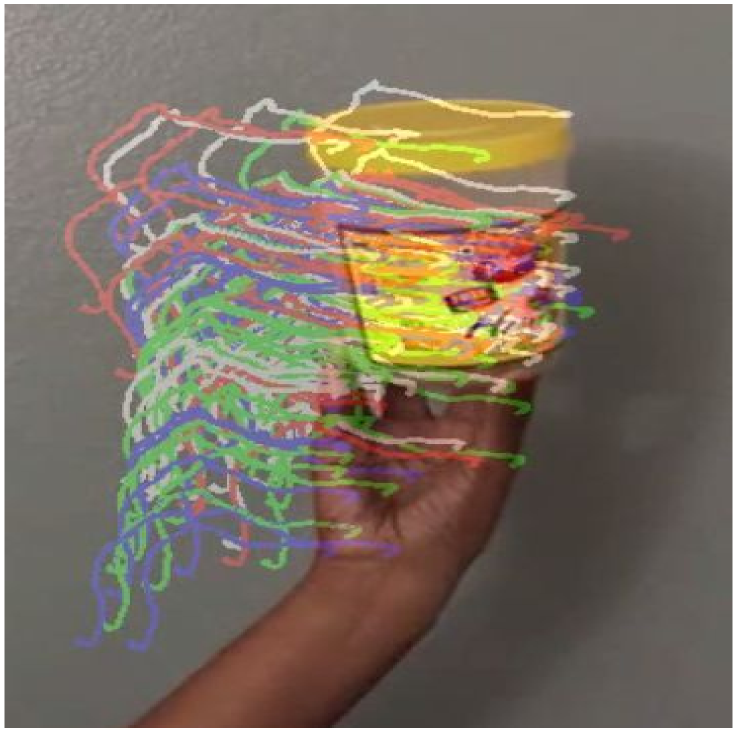


Using **SIFT**

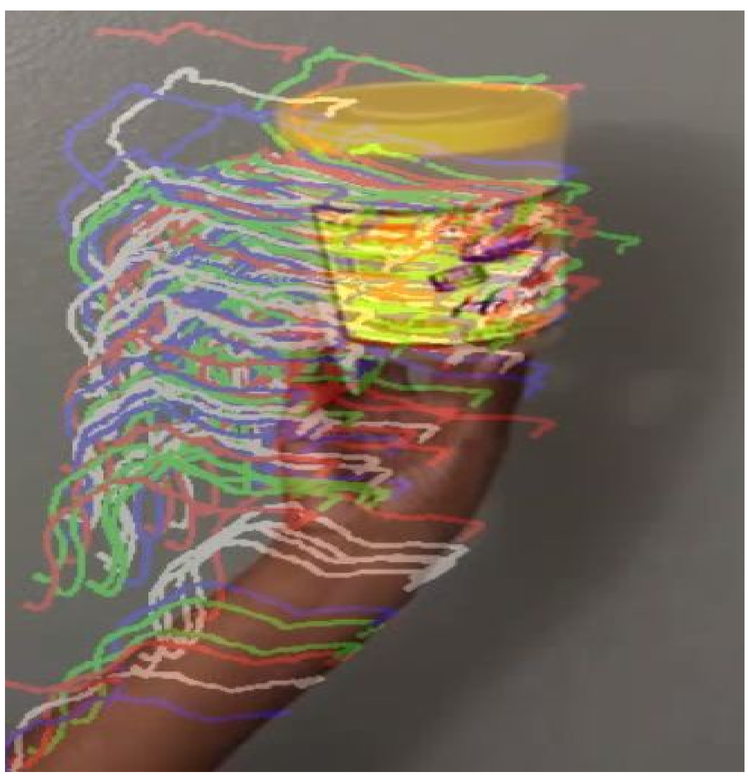


**2. A part of results of using KLT Optical Flow,**

Using **SURF**



Using **SIFT**



**Motion Estimation**

Execution time for each frame using SURF and KLT optical flow = 0.15sec

Execution time for each frame using SIFT and KLT optical flow = 0.16 sec

Execution time for each frame using SURF and Flann Matching = 0.56 sec

Execution time for each frame using SIFT and Flann Matching = 1.28 sec

The output of drawing lines between features detected in each frame is continuous using KLT optical flow whereas it is not continuous using features descriptors and descriptor matching as number of descriptors computed for each frame may not of same count leading to the discontinuity.

**Conclusions:**

The SURF feature detector requires less execution time compared to SIFT ( SURF is 3 times faster than SIFT) and has comparable performance as that of SIFT. Hence SURF is a fast approximation of SIFT algorithm.

Motion Estimation using KLT Optical flow method is better than one using Feature Descriptors and FLANN matcher in terms of elapsed time and accuracy. The lines flow between consecutive frames is continuous in the approach of optical flow.

The amount of background light also affected the performance. When the background light was too bright, the KLT optical flow performed better than Feature Descriptor method. The number of keypoints with good tracking lines was much less in Feature Descriptor method.

Future work may include using other descriptors like ORB, BRISK etc for keypoints detection and any other descriptor match functions like BFMatcher etc. BFMatcher generally took more time then FLANN matcher. For special case when the angle of rotation is proportional to 90 degrees, ORB and SURF outperforms SIFT and in the noisy images, ORB and SIFT show almost similar performances. In SURF, SIFT and FAST keypoint detectors are distributed over the image while in ORB, the features are mostly concentrated in objects at the center of the image.

**References:**


[1] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” International Journal of Computer Vision, vol. 60, issue 2, pages 91-110, Nov. 2004.

[2] Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision.

[3] C. Tomasi and T. Kanade, “Detection and Tracking of Point Features,” Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[4] https://dsp.stackexchange.com/questions/10423/why-do-we-use-keypoint-descriptors

[5] http://www.mee.tcd.ie/~sigmedia/pmwiki/uploads/Teaching.4S1b/handout9_4s1.pdf

[6] https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf