

Chapter 2

Moving Object Detection and Tracking in Videos

Abstract This chapter provides four sections. The first section introduces the moving object D&T infrastructure and basis of some methods for object detection and tracking (D&T) in videos. In object D&T applications, there is manual or automatic D&T process. Also, the image features, such as color, shape, texture, contours, and motion can be used to track the moving object(s) in videos. The detailed information for moving object detection and well-known trackers are presented in this section as well. In second section, the background subtraction (BS) method and its applications are given in details. The third section declares the details for Mean-shift (MS), Mean-shift filtering (MSF), and continuously adaptive Mean-shift (CMS or CAMShift) methods and their applications. In fourth section, the details for the optical flow (OF), the corner detection through feature points, and OF-based trackers are given in details.

Keywords Background subtraction · Mean-shift · CAMShift · Mean-shift filtering · Optical flow

2.1 Introduction

In video processing, a video can be represented with some hierarchical structure units, such as scene, shot and frame. Also, video frame is the lowest level in the hierarchical structure. The content-based video browsing and retrieval, video-content analysis use these structure units. In video retrieval, generally, video applications must first partition a given video sequence into video shots. A video shot is defined as an image or video frame sequence that presents continuous action. The frames in a video shot are captured from a single operation of one camera. The complete video sequence is generally formed by joining two or more video shots [55, 56].

According to Koprinska and Carrato [56], there are two basic types of video shot transitions, namely abrupt and gradual. Abrupt transitions (i.e., cuts) are sim-

plest form, which occur in a single frame when stopping and restarting the camera. Although many kinds of cinematic effects could be applied to artificially combine two or more video shots. Therefore, the fades and dissolves are most often used to create gradual transitions. A slow decrease in brightness resulting in a black frame is a fade-out. In addition, a fade-in is a gradual increase in intensity starting from a black image. However, dissolves show one image superimposed on the other as the frames of the first video shot get dimmer and those of the second one get brighter [56].

In Camara-Chavez et al. study [55], they expressed that video shots can be effectively considered as the smallest indexing unit where no change in the scene content can be perceived. In addition, the higher level concepts are often constructed by combining and analyzing the inter- and intra-shot relationships. For a video or multimedia indexing, or editing application, each video shot can be generally represented by key frames and indexed according to the spatial and temporal features [55, 57]. In the literature, several content-based retrieval systems for organizing and managing video databases have been already proposed [56]. In Fig. 2.1, a schematic for content-based retrieval of video databases is shown. In this schematic, one can see that video shots cover key frames, and above mentioned temporal and spatial features are extracted from these video shots as well. In addition, the end-users may interact with such a retrieval system and those interactions lead to the indexing and annotation processes via browse, search, edit, or analyze operations.

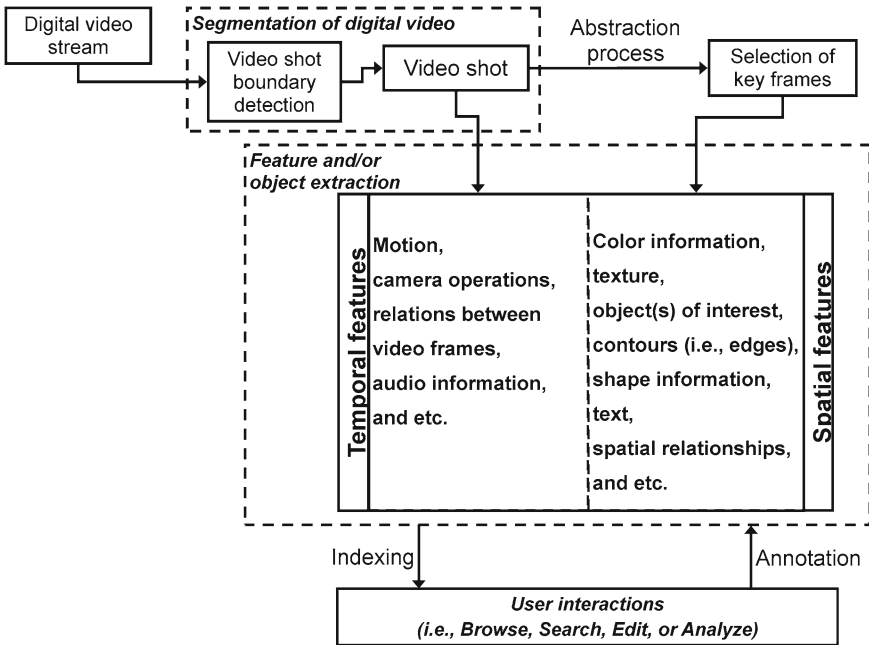


Fig. 2.1 The schematic for content-based retrieval of video databases

Human and computer vision work similar in terms of functionality, but have not exactly same functions and results [57, 58]. In this frame, the essentials of visual surveillance systems based on the basic elements should be considered. A camera is a device that records and stores (still) images from a given scene, also it is the basic sensing element [57]. Usually, the camera operations explicitly reflects how the attention of the viewer (i.e., human) should be directed [56]. Sensing with camera is the first step for a object D&T process of a visual surveillance system. A movie or video (i.e., captured by a camera) continuously takes some video frames per second as long as the user's choices (i.e., the length of the video). Generally, digital signal and image processing are the essential parts or levels for digital video processing. The object detection process affects on the object tracking and classification processes in video-content analysis, and video information retrieval via digital video processing [57].

In object D&T applications, manual D&T of object(s) is a tedious and exhausting task. Therefore, the experts in computer vision area studied for long periods of time on semiautomatic and automatic D&T techniques. These D&T techniques often involve maintaining a model, which is related to the spatial relationship between the various features [59]. In the literature, some image features, such as color, shape, texture, contours (e.g., edges), and motion (i.e., trajectory and spatial relationship) can be used to track the moving object(s) in videos. One can see that these features are given as spatial features in Fig. 2.1.

Video segmentation has two major types which are spatial and temporal segmentation. The spatial segmentation is based on the approach of digital image segmentation. Also, the temporal segmentation is constructed by time-based, meaningful, and manageable segments (i.e., video shots) in videos. According to Koprinska and Carrato [56], the temporal video segmentation is the first step toward automatic annotation of digital video sequences, and thus, its goal is to divide the video stream into a set of video shots that are used as basic elements for indexing. Therefore, each video shot is then represented by selecting key frames and indexed by extracting spatial and temporal features. The video retrieval process is based on the similarity between feature vector of the query and already stored video features [56].

Digital image segmentation is generally used to partition an image as bi-level or higher level into different regions that they belong to given image or video frame. This partition process can be either locally or globally. Semiautomatic image segmentation involves end-user interactions to separate the interested object(s) from background that the object is involved by the region of interest (ROI). Automatic image segmentation is similar to semiautomatic one, but it aims to separate and identify the object(s) in given ROI or in whole image that it works without end-user's intervention. The identification is based on the accurate boundaries of the object(s).

Tracking of a moving object over time is a challenging issue on video processing. The researchers developed a lot of video processing software to detect the position of an object in each image (i.e., video frame) in given sequence, and hence a temporal sequence of coordinates is determined [60]. If one realize this process for each image in given sequence and simply concatenate the positions of given object, thus the tracking is often considered as successfully accomplished. According to the

textbook of Moeslund [60], the above mentioned approach, however, not considered tracking since each detection is done independently of all other detections. The main reason is that there is no temporal information in tracking process. The points in a coordinate system that are traveled by an object in a time scale are considered as the trajectory of this moving object. This approach can be extended to the states of object. These states are often stored in a state vector, where each entry in this vector contains the value of a certain parameter at a particular time [60]. Therefore, a general form of a moving object tracking process turns into an update process of previous states. The entries of state vector could be some features, such as position, velocity, acceleration, size, shape, color, texture and etc.

According to the study of Liu et al. [4], in distribution-based object D&T approach, Background Subtraction (BS) is the most popular method to detect moving object(s). The main idea of this approach is to estimate an appropriate representation (i.e., background image model) of the given scene based on pixel distribution. Also, the object(s) in the current video frame can be detected by subtracting the current video frame with the background model.

In addition, Liu et al. [4] expressed that the Optical Flow (OF) is the most widely used method in orientation-based object D&T approach. The OF approach approximates the moving object motion by estimating vectors originating or terminating at pixels in image (i.e., video frame) sequences. Also, the velocity field is represented by OF, which warps one image into another high dimensional feature space. The motion detection methods based on OF can accurately detect motion in the direction of intensity gradient [9, 10]. However, Liu et al. [4] drew attention to issue of motion that the motion is tangential to the intensity gradient cannot be well represented by the feature map. Furthermore, the illumination changes affect badly on OF-based methods.

The contour-based object D&T approaches also are not covered in our study. The main reason is that these methods cannot handle fast moving object very well, also they are computationally expensive and insensitive to illumination changes [4, 6–8].

The approaches based on the color probability distribution or color-clustering deal often with dynamical changes of color probability distribution (CPD). In this frame, in order to track colored object(s) in given video frame sequence, the color image data has to be represented as a probability distribution [57]. Generally, color distributions derived from video frame sequences change over time, also the object D&T method has to be modified to adopt dynamically to the probability distribution. Therefore, such methods use color histograms to track moving object(s). A good example to this approach is CAMShift tracker that it is based on MS color clustering approach. This color clustering (i.e., spatial segmentation) is also based on MS filtering (MSF) procedure that is a straightforward extension of the discontinuity preserving smoothing algorithm [36, 57]. This kind of object D&T process has lower computation cost than the approaches based on graph-cut or active contour models as well.

In our study, detailed information is given in Sect. 2.2 for background subtraction (BS) method, in Sect. 2.3 for Mean-shift (MS), Mean-shift filtering (MSF) and continuously adaptive Mean-shift (CMS or CAMShift) methods, in Sect. 2.4 for

optical flow (OF) method and its variants as Horn–Schunck (HS) (i.e., Dense OF) and Lucas–Kanade (LK) (i.e., Sparse OF) techniques in this book, respectively. These methods are used for both of object D&T steps in our ViCamPEv software.

2.2 Background Subtraction

The detection of interesting foreground object from a video sequence provides a classification of the pixels into either foreground or background [61]. A scene in object detection process can be usually represented with a model called background model. The related algorithm (or method) finds deviations from the model for each incoming frame. Note that, the former form of this method is called sometimes frame differencing. This process is usually referred as the background subtraction [17, 30]. When the scene is stationary or gradually evolving, then the foreground detection can be solved conveniently with many traditional BS algorithms [61]. The statistical modeling techniques [12] and their variants [13] are often used in the literature [61]. The performance of these methods or techniques deteriorates when the scene involves dynamic elements or occluded fronts, such as waving trees, flocks of birds, rippling water, fog, or smoke, etc.

Common BS techniques were reviewed in Benezeth et al. study [62]. According to Benezeth et al. [62], the principle of BS methods can be summarized by the following rule,

$$\lambda_t(s) = \begin{cases} 1, & \text{if } m(I_{s,t}, B_s) > \tau_H \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

where λ_t is the motion label field at time t and a function of $s(x, y)$ spatial location (also called motion mask), m is a distance between $I_{s,t}$ the video frame at time t at pixel s and B_s the background at pixel s ; and τ_H is a threshold. The main differences between most BS methods are how well B_s is modeled and which distance metric m is being used (e.g., Euclidean, Manhattan, Mahalanobis, and etc.) [17, 62]. There are four main steps in a BS algorithm which are explained in Cheung and Kamath's study [63], namely the preprocessing, background modeling, foreground detection, and data validation. According to Cheung and Kamath's study [63], preprocessing step involves some simple image processing tasks, which change the raw input video sequence into a format that is used in subsequent steps. In background modeling step, the new video frame can be used in the calculation and updating process of a background model. This model provides a statistical description of the entire background scene. This scene may be static or dynamic [57]. In the foreground detection step, some pixels in given video frame, which are not explained enough by given background model [32], are defined as a binary candidate foreground mask [57, 63].

In this frame, the background modeling techniques can be classified into two broad categories: nonrecursive and recursive techniques. For nonrecursive techniques, representative work includes frame differencing, median filter, linear predictive filter, and nonparametric model. A nonrecursive technique uses a sliding-window approach

for background estimation. This approach stores a buffer of the previous video frames. Therefore, it estimates the background image based on the temporal variation of each pixel within the buffer. These techniques are highly adaptive as they do not depend on the history beyond those frames stored in the buffer [63]. The recursive techniques are frequently based on the approximated median filter, Kalman filter, and mixture of Gaussian (MoG) [57]. Recursive techniques do not maintain a buffer for background estimation. Also, they recursively update a single background model based on each input video frame. As a result, input frames from distant past could have an effect on the current background model. According to the study of Cheung and Kamath [63], compared with nonrecursive techniques, the recursive techniques require less storage. In addition, any error in the background model of these techniques can linger for a much longer period of time. In Fig. 2.2, the schematic for a general object D&T system based on the recursive BS is shown.

In Fig. 2.2, there are five steps: preprocessing, BS with updating, foreground detection, post-processing, and tracking with foreground mask. The preprocessing and BS with updating (i.e., background modeling) steps are similar to the steps which are explained in Cheung and Kamath's study [63]. In post-processing step, some corrections are made on given binary image via connected component analysis (CCA), and etc. In tracking with foreground mask step (i.e., Step V in Fig. 2.2), the tracking process is achieved by spatio-temporal information about moving object. At decision point between Step IV and Step V in Fig. 2.2, the algorithm that works on underlying system looks up the spatial and temporal coherency for moving object; if given object is labeled as stationary in a certain number of video frames, then this object will be embedded in the background model as a part of the model, and the background model can be updated immediately with this new information. However,

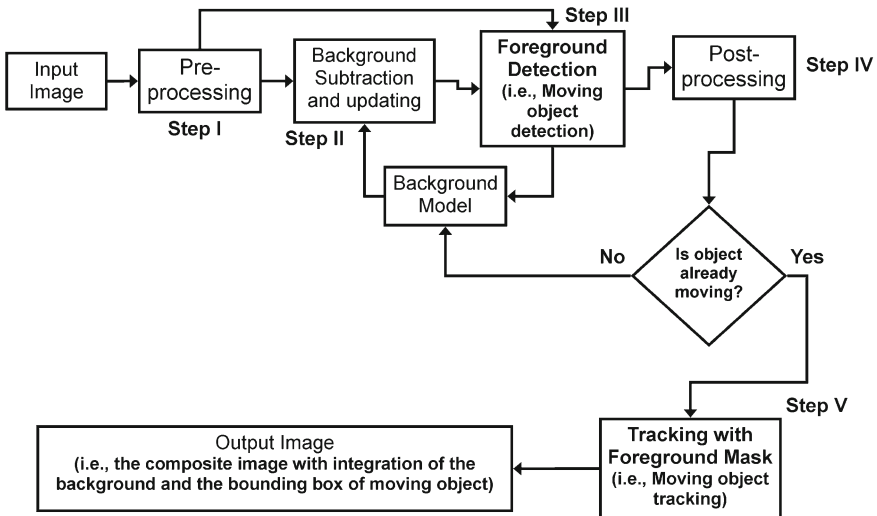


Fig. 2.2 The schematic for a general object D&T system based on the recursive BS

if the object is already moveable, then it is tracked as a foreground mask. This foreground mask is frequently a binary image (i.e., black and white image) and also its white regions are usually treated as the moving objects by BS algorithm.

In the literature, many approaches for automatically adapting a background model to dynamic scene variations are proposed. Detailed information for the aforementioned techniques can be found in the literature [30, 62, 63] as well. Some implementations of recursive BS are used in our ViCamPEv software.

2.2.1 Literature Review

For foreground detection in dynamic scenes, there are two main categories of relevant approaches, namely pixel level and region level [61]. In the pixel-level models, the scene model has a probability density function (PDF) for each pixel separately. According to Zivkovic and van der Heijden [13], a pixel from a new image is considered to be a background pixel, if its new value is well described by its density function. The simplest mode is often given for a static scene that it could be just an image of the scene without the intruding objects [13]. Furthermore, the variances of the pixel intensity levels from the image can vary from pixel to pixel. In the study of Mittal and Paragios [11], they used an elaborated adaptive kernel density estimation scheme to build a nonparametric model of color and optical flow (OF) at each pixel. According to Wu and Peng's study [61], when the same motions are observed many times in a certain number of frames, then the method of Mittal and Paragios [11] provides good detection results. The nonparametric density estimates also lead to flexible models. In addition, the kernel density estimate was proposed for BS in the literature [15], but there is a problem with kernel estimates that the choice of the kernel size is considered as fixed. This problem can be overcome by the use of variable-size kernels [64]. In the study of Wang and Suter [65], they presented a concept named 'sample consensus'. This concept defined how many times the current pixel agreed with previous samples at that pixel site, and thus, the foreground and background are separated by thresholding each consensus value.

On the other hand, the region-level methods are based on the relationship between pixels. In the study of Sheikh and Shah [66], they proposed a kernel density estimation to model the full background as a single distribution. In their study, the pixel locations were unified into the distribution, eliminating one distribution per pixel. Their experimental results show that the appealing performance of Sheikh and Shah's method. In the study of Zhong and Sclaroff [67], an autoregressive moving average model was employed to model the dynamic textured background, and exploited a robust Kalman filter to estimate the intrinsic appearance of dynamic texture as well as the foreground regions. In addition, Dalley et al. [68] introduced a generalization of the MoG model to handle dynamic textures. According to the study of Wu and Peng [61], they treated the image generation process as arising from a mixture of components consisting of a Gaussian distribution in color and some spatial distribution. Experiments validated the correctness and effectiveness of their algorithm [68].

In the literature, there are a lot of studies that they issued some variants of BS method. Mandellos et al. [69] presented an innovative system for detecting and extracting vehicles in traffic surveillance scenes. Their system covers locating moving objects present in complex road scenes by implementing an advanced BS methodology. In their study, a histogram-based filtering procedure was concerned by the innovation that this procedure collects scatter background information carried in a series of frames, at pixel level, generating reliable instances of the actual background. A background instance on demand under any traffic conditions was reconstructed by the proposed algorithm. According to Mandellos et al. study [69], the background reconstruction algorithm demonstrated a rather robust performance in various operating conditions including unstable lighting, different view-angles, and congestion.

In the study of Spagnolo et al. [70], they addressed the problem of moving object segmentation using BS. They proposed a reliable foreground segmentation algorithm that combines temporal image analysis with a reference background image. In their study, a new approach for background adaptation to changes in illumination was presented. All the pixels in the image, even those covered by foreground objects that they are continuously updated in the background model. The experimental results of their study demonstrated the effectiveness of the proposed algorithm when applied in different outdoor and indoor environments.

In the study of Zhang and Ding [71], a tracking algorithm based on adaptive BS about the video D&T moving objects was presented. In first stage, they used a median filter to achieve the background image of the video and denoise the sequence of video. In second stage, they used adaptive BS algorithm to detect and track the moving objects. Adaptive background updating was also realized by the study of Zhang and Ding [71], finally, they improved the accuracy of tracking through open operation. The simulation results of their study show that the adaptive BS is useful in both D&T moving objects, and BS algorithm runs more quickly.

In the study of Shoushtarian and Bez [72], they presented and compared three dynamic BS algorithm for color images. The performances of these algorithms defined as ‘Selective Update using Temporal Averaging’, ‘Selective Update using Non-foreground Pixels of the Input Image’, and ‘Selective Update using Temporal Median’ are only different for background pixels. Then, they used an invariant color filter and a suitable motion tracking technique, an object-level classification was also offered that recognizes the behaviors of all foreground blobs. Their approach, which selectively excludes foreground blobs from the background frames, was included in all three methods. They showed that the ‘Selective Update using Temporal Median’ produces the correct background image for each input frame. The third algorithm operates in unconstrained outdoor and indoor scenes. Also, the efficiency of the new algorithm was confirmed by the results obtained on a number of image sequences.

In the study of Magee [73], a vehicle tracking algorithm was presented based on the combination of a novel per-pixel background model and a set of foreground models of object size, position, velocity, and color distribution. The background model is based on the Gaussian mixture (GM). According to study of Magee [73], each pixel in the scene was explained as either background, belonging to a foreground object, or as noise. A projective ground-plane transform was used within the foreground

model to strengthen object size and velocity consistency assumptions. In the study, a learned model of typical road travel direction and speed was used to provide a prior estimate of object velocity, which is used to initialize the velocity model for each of the foreground objects. In the experimental results, their system was worked at near video frame rate (i.e., greater than 20 fps) on modest hardware and is robust assuming sufficient image resolution is available and vehicle sizes do not greatly exceed the priors on object size used in object initialization.

In El Maadi and Maldague's study [74], a framework was proposed to detect, track, and classify both pedestrians and vehicles in realistic scenarios using a stationary infrared camera. In addition, a novel dynamic BS technique to robustly adapt detection to illumination changes in outdoor scenes was proposed. Their experimental results show that combining results with edge detection enables to reduce considerably false alarms (FA) while this reinforces also tracking efficiency. El Maadi and Maldague declared that their proposed system was implemented and tested successfully in various environmental conditions.

In Davis and Sharma's study [75], they presented a new BS technique fusing contours from thermal and visible imagery for persistent object detection in urban settings. Statistical BS in the thermal domain was used to identify the initial ROI. Also, color and intensity information were used within these areas to obtain the corresponding ROIs in the visible domain. Within each region, input and background gradient information were combined to form a contour saliency map (CSM). The binary contour fragments were obtained from corresponding CSMs that they are then fused into a single image. In their study, an A^* path-constrained search (i.e., a search algorithm that is widely used in pathfinding and graph traversal) along watershed boundaries of the ROIs was used to complete and close any broken segments in the fused contour image. At the end, the contour image was flood-filled to produce silhouettes. The results of their approach were evaluated quantitatively and compared with other low- and high-level fusion techniques using manually segmented data.

2.3 Mean-Shift and Continuously Adaptive Mean-Shift

In the literature, Mean-shift (MS) approach is a clustering approach in image segmentation. MS was originally proposed by Comaniciu and Meer [76] to find clusters in the joint spatial-color space. In the literature, MS algorithm is detailed in some studies [17, 30, 36, 76–79]. MS is susceptible to fall into local maxima in case of clutter or occlusion [79]. Nummiaro et al. [80] declared that MS-based trackers easily fail in tracking rapid moving objects. They cannot recover from the possible failures. Furthermore, these trackers' efficiency is important against robustness. In addition, they cannot deal with multimodal (i.e., for scenes with multiobject) problems [17, 81, 82]. Continuously adaptive Mean-shift (CMS or CAMShift) is a tracking method. CMS is a modified form of MS method. In CMS, MS algorithm is modified to deal with dynamically changing color probability distributions (CPDs) derived from video frame sequences. Color histograms were used in Bradski's study [34] via new

algorithm is called CMS. In a single image (or in a single frame of a video sequence), the CMS process is iterated until convergence criterion is met [17]. When the tracked object's color does not change, the MS-based CMS trackers are quite robust. However, if similar colors appear in the background, then MS-based CMS trackers easily fail in tracking. The Coupled CMS algorithm was demonstrated in a real-time head tracking application [34, 35]. The algorithm of a general CMS tracker is used as a part of Open Source Computer Vision library (OpenCV) [77, 78]. Furthermore, the CMS tracker tracks usually a specified object, which is defined or detected by system [17]. Similar implementation of CMS tracker is used in our ViCamPEv software.

2.3.1 Mean-Shift and Mean-Shift Filtering

The segmentation procedure based on the Mean-shift (MS) analysis is used to analyze complex multimodal feature space and identification of feature clusters. This procedure is iterative and used to seek the mode of a density function presented by local samples. Furthermore, its approach is called nonparametric, because a complete density is being modeled [17, 36, 78, 83, 84]. The MS clustering is a method to cluster an image by associating each pixel with a peak of the image's probability density, and also, it is provided that this procedure is a quadratic bound maximization both for stationary and evolving sample sets [85]. For MS clustering, its ROI size and shape parameters are only free parameters on MS process, i.e., the multivariate density kernel estimator [57].

The MS algorithm is initialized with a large number of hypothesized cluster centers randomly chosen from the data of the given image. The algorithm aims at finding of nearest stationary point of underlying density function of data [30]. The peak in the local density is computed by first defining a window in the neighborhood of the pixel. Also, its utility is that the detecting the modes of the density is easy by using MS process. Therefore, the mean of the pixel that lie within the window can be calculated. This window is then shifted to the mean [86]. For this purpose, each cluster center is moved to the mean of the data lying inside the multidimensional window centered on the cluster center [30]. Until convergence, the similar steps are repeated. The outcome of the MS process is only controlled by the kernel size (i.e., bandwidth). Thence, MS requires less manual intervention compared to other clustering algorithms. However, too large or small bandwidth may lead under- or over-segmentation problems. Zhou et al. [86] discussed about the integration of different algorithms with MS in order to find an optimal solution that this integration is used to effectively handle segmentation problems. In MS clustering procedure, the algorithm builds up a vector that is defined by the old and the new cluster centers, which is called MS vector. It is computed iteratively until the cluster centers do not change their positions. Also, some cluster may get merged during the iterations of MS process [30, 36, 57, 76].

The mean-shift filtering (MSF) is based on a smooth continuous nonparametric model. In the literature, a well-known noniterative discontinuity preserving

smoothing technique is the bilateral filtering (BF). According to Comaniciu and Meer's studies [36, 76], the use of local information [76] is the essential difference between the BF- and the MSF-based smoothing. In MSF, the spatial coordinates of each pixel are adjusted along with its color values (i.e., joint domain of color and location). This joint domain of color and location is based on $[l, u, v, x, y]$ space, where (l, u, v) represents the color and (x, y) represents the spatial location [30]. Therefore, the pixel migrates more quickly toward other pixels with similar colors. It can later be used for clustering and segmentation [83]. The image segmentation based on MSF procedure is a straightforward extension of the discontinuity preserving smoothing algorithm. In the process, each pixel associates with a significant mode of the joint domain density located in its neighborhood [76].

The MS vector is derived by estimating the density gradient that the vector always points toward the direction of maximum increase in density. The density modes in the feature space (i.e., local maxima) can thus be located by computing the MS vector [87]. In addition, the smoothing through replacing the pixel in the center of a window by the (weighted) average of the pixels in the window indiscriminately blurs the image [76]. However, it removes not only the noise, but also the salient information. The MS-based discontinuity preserving smoothing algorithm is called as the mean-shift filtering (MSF). Also, the information beyond the individual windows is taken into account in the image smoothed by MSF [76].

In the d -dimensional Euclidean space \mathbf{R}^d , given n data points $x_i, i = 1, 2, \dots, n$, the kernel density estimator of point x with kernel $K(x)$ and bandwidth h is given by [76, 87]

$$\hat{f}_{h,K}(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right). \quad (2.2)$$

In the mathematical background, the radially symmetric kernels are usually used which satisfy

$$K(x) = c_{k,d} k\left(\|x\|^2\right) \quad (2.3)$$

where $c_{k,d}$ is the normalization constant, and $k(x)$ is called the profile of the kernel [76, 87]. Also, $x \in \mathbf{R}^d$ is a point in the d -dimensional feature space, and $\|\cdot\|$ is a norm. The density estimator in Eq. (2.2) can be rewritten as [87]

$$\hat{f}_{h,K}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{x-x_i}{h}\right\|^2\right). \quad (2.4)$$

According to Szeliski [83], starting at some guess for a local maximum, y_j , which can be a random input data point x_i , MS clustering computes the gradient of the density estimate $\hat{f}_{h,K}(x)$ at y_j and takes an uphill step in that direction [83]. This approach is a variant of multiple restart gradient descent. Therefore, the gradient of $\hat{f}_{h,K}(x)$ can be expressed as [76, 87]

$$\nabla \hat{f}_{h,K}(x) = \hat{f}_{h,Q}(x) \frac{2c_{k,d}}{h^2 c_{q,d}} m_{h,Q}(x) \quad (2.5)$$

where $q(x) - k'(x)$, and $k'(x)$ is the first derivative of $k(x)$. The Q is another kernel that its profile is q . In this frame, the MS vector $m_{h,Q}(x)$ can be expressed as [87]

$$m_{h,Q}(x) = \frac{\sum_{i=1}^n x_i q \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)}{\sum_{i=1}^n q \left(\left\| \frac{x-x_i}{h} \right\|^2 \right)} - x. \quad (2.6)$$

As Szeliski [83] discussed in his textbook, the MS vector acts as the difference between the weighted mean of the neighbors x_i around x and the current value of x . The kernels are adjusted accordingly to location and color that they may have different scales. A multivariate normal kernel to be optimal one for the MS procedure that is given as [76]

$$K(x) = (2\pi)^{-d/2} \exp \left(-\frac{1}{2} \|x\|^2 \right). \quad (2.7)$$

The Epanechnikov kernel is the shadow of the uniform kernel, i.e., the d -dimensional unit sphere, while the normal kernel and its shadow have the same expression [76]. The CMS trackers use MS segmentation that it is based on MSF.

Denote by $\{y_j\}_{j=1,2,\dots}$ the sequence of successive locations of the kernel Q . Let x_i and $z_i, i = 1, \dots, n$, be the d -dimensional input and filtered image pixels in the joint spatial-range domain, respectively. This joint spatial-range domain $d[x_p, y_p, l, u, v]$ is a five dimensional vector, where (x_p, y_p) is the pixel location and (l, u, v) are the values of color components. Comaniciu and Meer [76] used the $CIE L^*u^*v^*$ as the perceptually uniform color space. They employed $CIE L^*u^*v^*$ motivated by a linear mapping property. In addition, there is no clear advantage between using $CIE L^*u^*v^*$ or $CIE L^*a^*b^*$ color space. The $CIE L^*a^*b^*$ is a three-dimensional model that each color is treated as a point in a three-dimensional space, where the difference between two colors is based on the Euclidean distance. It has a color-opponent space with dimension L for lightness, and also, a and b for the color-opponent dimensions. In this context, the MSF procedure in Table 2.1 can be also applied to each pixel as given below [76, 87].

In Table 2.1, the superscripts s and r denote the spatial and range components of given vector, respectively. Comaniciu and Meer [76] discussed that the filtered data at the spatial location x_i^s will have the range component of the point of convergence $y_{i,c}^r$. At the end of the MS clustering (also MSF) process, all data points was visited by the MS procedure converging to the same mode, which forms a cluster of arbitrary shape [87].

Table 2.1 The MSF procedure from the study of Comaniciu and Meer [76]

Step 1.	Initialize $j = 1$ and $y_{i,1} = x_i$
Step 2.	Compute $y_{i,j+1}$,
	$y_{j+1} = y_j + m_{h,q} = \frac{\sum_{i=1}^n x_i q \left(\left\ \frac{y_j - x_i}{h} \right\ ^2 \right)}{\sum_{i=1}^n q \left(\left\ \frac{y_j - x_i}{h} \right\ ^2 \right)}, j = 1, 2, \dots$
	until convergence, $y = y_{i,c}$,
Step 3.	Assign $z_i = (x_i^s, y_{i,c}^r)$

2.3.2 Continuously Adaptive Mean-Shift

The MS segmentation is based on the MSF procedure that it is based on MS clustering. The advantages of MS segmentation is its modularity that the control of segmentation output is very simple. Comaniciu and Meer [76] introduced the basis of MS segmentation in their study that x_i and $z_i, i = 1, 2, \dots, n$, are given as the d -dimensional input and filtered image pixels in the joint spatial-range domain and Ψ_i as the label of the i th pixel in the segmented image, respectively. Therefore, the MS segmentation procedure in Table 2.2 can be also applied to each pixel [76, 87].

The essential idea is that an initial window (i.e., kernel) is placed over a two-dimensional array of data points and is successively recentered over the mode (or local peak) of its data distribution until convergence [77]. As mentioned before, the CMS or CAMShift is a tracking method that is a modified form of MS method. The MS algorithm operates on color probability distributions (CPDs), and CMS is a modified form of MS to deal with dynamical changes of CPDs [57]. In the literature, some color spaces were used to detect and track the moving object(s). In color image segmentation or color clustering, the most significant point is how to measure the difference between two or more colors. Also, *RGB* stands for Red (*R*), Green (*G*), and Blue (*B*) colors. The *HSV* stands for Hue (*H*), Saturation (*S*), and Value (*V*). In addition, *HSV* has an alternative name as *HSL/I* that it has hue, saturation, and

Table 2.2 The MS segmentation procedure from the study of Comaniciu and Meer [76]

Step 1.	Run the MSF procedure for the image and store all the information about d -dimensional convergence point in $z_i, z_i = y_{i,c}$,
Step 2.	Delineate in the joint domain the clusters $\{C_p\}_{p=1,2,\dots,m}$ by grouping together all z_i , which are closer than h_s , in the spatial domain and h_r , in the range domain, by this way, the basins of attraction of the corresponding convergence points are concatenated.
Step 3.	For each $i = 1, \dots, n$, assign $\Psi_i = \{p \mid z_i \in C_p\}$.
Step 4.	As an optional choice: Eliminate spatial regions containing less than M pixels.

lightness/intensity components. The *HSV* is a most common cylindrical-coordinate representation of points, which are given in a *RGB* color space model.

In the literature, the *HSV* color space is used to segment, detect, and track the moving object by CMS trackers. To track colored objects in video frame sequences, the color image data has to be represented as a probability distribution [34, 35, 57]. The main reason is that the computational complexity of *HSV* is lower than *CIE L*a*b** color space. In addition, the *HSV* color space has a better quantization than *CIE L*a*b** color space, and also, the better quantization is to cause a better segmentation, but the *HSV* color space has a worse computational complexity than *RGB* color space. Kerminen and Gabbouj [88] compared three different color spaces with each other, which are *RGB*, *HSV*, and *CIE L*a*b** color spaces. The reader may refer for detailed information to the study of Kerminen and Gabbouj [88].

To accomplish the tracking moving object, Bradski used color histograms in his study [34, 57]. Color distributions derived from video image sequences change over time, so the MS algorithm has to be modified to adapt dynamically to the probability distribution it is tracking. The novel algorithm in Bradski's study (i.e., CMS) meets all these requirements. In a single image, the CMS (or CAMShift) process is iterated until convergence or until an upper bound on the number of iterations is reached. When the tracked target object's color does not change, the MS-based CMS trackers are quite robust and stable. However, they are easily distracted when similar colors appear in the background.

The Coupled CMS algorithm (as given in the study of Bradski [34]) is demonstrated in a real-time head tracking application that it is a part of the Intel OpenCV Computer Vision library [77, 78]. The Coupled CMS is reviewed in François's study [35] as well. The reader may refer to the reference [34] and [35] for some implementations of the algorithm in face tracking. In addition, the face tracking is used to control various interactive programs via head movements [57]. The head's color model is actually a skin color tone model in *HSI* (i.e., *HSV*) color space. The Hue component of *HSI* for color images gives relatively best results for skin color tone-based tracking. A universal framework for the distributed implementation of algorithms called software architecture for *immersipresence* is used by François [35] as well.

A system design involving a CAMShift tracker is presented by François [35] as a black-box approach based on the CAMShift tracker via an instance of *CvCamShiftTracker* class of OpenCV library. In addition, OpenCV and the software architecture for *immersipresence* are used to design and implement a CAMShift-based tracking system in the study of François [35]. For better understanding, an exemplary depiction of the design is given in Fig. 2.3 in which OpenCV's CAMShift tracker [57, 77, 78] is taken as basis for the design of general object D&T system based on a CAMShift tracker.

In Fig. 2.3, an input image (or the first frame of given video) in *RGB* color space is converted at Step I to *HSV* color space. Then, the histogram computation according to backprojection can be calculated in Step II. Also, in Step III, CAMShift tracker identifies the moving object and describes the new bounding box properties for object D&T such as box size and orientation. Thus, the last known bounding box can be updated by using this way. This is an iterative procedure. In Step IV, the set of

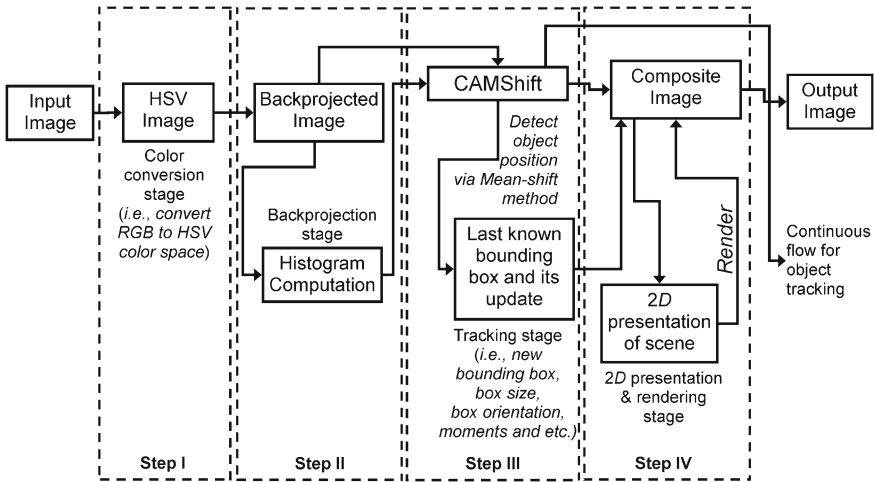


Fig. 2.3 The schematic for a general object D&T system based on a CAMShift tracker (figure adapted from the study of Karasulu [57])

bounding box, detected boundaries and segmented object are merged together on to original background image and rendered as a composite image (i.e., rendering using *2-Dimensional* scene information). The output image (or video frame) is based on this composite image, meanwhile, the flow of tracking for CAMShift continues until the video is finished.

2.3.3 Literature Review

In the literature, there are numerous studies related to the MS-based CMS (or CAMShift) trackers. In the study of Stern and Efros [89], they developed a procedure that adaptively switches color space models throughout the processing of a video. Also, they proposed a new performance measure for evaluating tracking algorithm. Their proposed methodology is used to find the optimal color space and color distribution models combination in the design of adaptive color tracking systems. Their color switching procedure was performed inside the framework of the CAMShift tracking algorithm. They combined a number of procedures to construct an enriched face tracking approach. At each iteration of the CAMShift algorithm, given image is converted into a probability image using the model of color distribution of the skin color being tracked. In the study of Li et al. [90], they proposed a novel approach for global target tracking based on MS technique. The proposed method represents the model and the candidate in terms of background- and color-weighted histogram, respectively, which can obtain precise object size adaptively with low computational complexity. Also, they implemented the MS procedure via

a coarse-to-fine way for global maximum seeking. This procedure was termed as adaptive pyramid MS, because it uses the pyramid analysis technique and can determine the pyramid level adaptively to decrease the number of iterations required to achieve convergence. The experimental results of the study of Li et al. [90] show that the proposed method can successfully cope with different situations such as camera motion, camera vibration, camera zoom and focus, high-speed moving object tracking, partial occlusions, target scale variations, etc.

Yuan et al. [91] proposed a new moving objects tracking algorithm, which combines improved local binary pattern texture and hue information to describe moving objects and adopts the idea of CAMShift algorithm. In order to reduce matching complexity on the premise of satisfying the accuracy, many kinds of local binary pattern and hue are cut down. According to Yuan et al. [91], the experiments show that the proposed algorithm can track effectively moving objects, can satisfy real-time and has better performance than others. In the study of Mazinan and Amir-Latifi [92], an improved convex kernel function was proposed to overcome the partial occlusion. Therefore, in order to improve the MS algorithm against the low saturation and also sudden light, changes are made from motion information of the desired sequence. By using both the color feature and the motion information simultaneously, the capability of the MS algorithm was correspondingly increased. In their study [92], by assuming a constant speed for the object, a robust estimator, i.e., the Kalman filter, was realized to solve the full occlusion problem. According to Mazinan and Amir-Latifi [92], the experimental results verified that the proposed method has an optimum performance in real-time object tracking, while the result of the original MS algorithm may be unsatisfied.

Jung and Han [93] proposed a hybrid approach of the two methods for text localization in complex images. An automatically constructed the texture classifier based on multilayer perceptron (MLP) can increase the recall rates for complex images with much less user intervention and no explicit feature extraction. The connected component-based filtering based on the geometry and shape information enhances the precision rates without affecting overall performance. Afterward, the time-consuming texture analysis for less relevant pixels was avoided by using CAMShift. According to Jung and Han [93], the experimental results show that the proposed hybrid approach leads to not only robust but also efficient text localization. In the study of Babu et al. [94], they presented a novel online adaptive object tracker based on fast learning radial basis function (RBF) networks. They have compared the proposed tracker against the MS tracker, which is known for robust object tracking in cluttered environment. Also, the pixel-based color features were used for developing the target object model in their study. In addition, two separate RBF networks were used, one of which is trained to maximize the classification accuracy of object pixels, while the other is trained for non-object pixels. The target was modeled using the posterior probability of object and non-object classes. Object localization was achieved by iteratively seeking the mode of the posterior probability of the pixels in each of the subsequent frames. Therefore, an adaptive learning procedure was presented to update the object model in order to tackle object appearance and illumination changes. According to Babu et al. [94], the superior performance

of the proposed tracker is illustrated with many complex video sequences, as compared against the popular color-based MS tracker. Therefore, the proposed tracker is suitable for real-time object tracking due to its low computational complexity.

Wang et al. [95] proposed a novel algorithm for tracking object in video sequence that it is called as CAMShift guided particle filter (CAMSGPF). CAMShift was incorporated into the probabilistic framework of particle filter (PF) as an optimization scheme for proposal distribution. CAMShift helps improve the sampling efficiency of particle filter in both position and scale space, also, it achieves better scale adaptation and can be applied in a simplified way without much loss in performance. According to Wang et al. [95], the CAMSGPF outperforms standard particle filter and MS embedded particle filter (MSEPF) based trackers in terms of both robustness and efficiency. Yin et al. [96] proposed an algorithm that combines CAMShift with PF using multiple cues. The effectiveness of particles was improved and the tracking window can change scale with the target adaptively because of the use of CAMShift. Meanwhile, an adaptive integration method was used to combine color information with motion information. Therefore, the problems can be solved with this way which are encountered in tracking an object with illumination variation and the background color clutter. In addition, an occlusion handler was proposed by Yin et al. [96] to handle the full occlusion for a long time.

In the study of González-Ortega et al. [97], a marker-free computer vision system for cognitive rehabilitation tests monitoring was presented. Their system monitors and analyzes the correct and incorrect realization of a set of psychomotricity exercises in which a hand has to touch a facial feature. Different human body parts were used D&T in this monitoring. Detection of eyes, nose, face, and hands was achieved with a set of classifiers built independently based on the AdaBoost algorithm. In their study, comparisons with other detection approaches, regarding performance, and applicability to the monitoring system were presented. Also, face and hands tracking was accomplished through the CAMShift algorithm. González-Ortega et al. [97] described the CAMShift algorithm that it is an adaptation of MS, which uses continuously adaptive probability distributions, i.e., distributions that can be recomputed for each frame, and with a search windows size adaptation. Also, they declared that CAMShift needs structural features of the tracking object, and it is robust to temporal variations of the features [34, 97]. In their study, the CAMShift algorithm was applied with independent and adaptive two-dimensional histograms of the chromaticity components of the *TSL* color space for the pixels inside these three regions. In *TSL* color space, a color is specified in terms of Tint (*T*), Saturation (*S*), and Luminance (*L*) values. It has the advantage of extracting a given color robustly while minimizing illumination influence. In their study, the *TSL* color space was selected after a study of five color spaces regarding skin color characterization. According to González-Ortega et al. [97], the experimental results show that their monitoring system was achieved a successful monitoring percentage.

Hu et al. [98] presented an enhanced MS tracking algorithm using joint spatial-color feature and a novel similarity measure function. The target image was modeled with the kernel density estimation and new similarity measure functions were developed using the expectation of the estimated kernel density. Also, two

similarity-based MS tracking algorithms were derived with these new similarity measure functions. In addition, the weighted-background information was added into the proposed tracking algorithm to enhance the robustness. The principal components of variance matrix were computed to update the orientation of the tracking object. The corresponding eigenvalues were used to monitor the scale of the object. According to Hu et al. [98], the experimental results show that the new similarity-based tracking algorithms can be implemented in real-time and are able to track the moving object with an automatic update of the orientation and scale changes.

2.4 Optical Flow

An optical flow (OF) is based on the idea that the brightness continuous for most points in the image [30, 57, 99], because of almost the same brightness is appeared on the neighboring points. The continuity equation for the optical term is given below [99],

$$\frac{\partial b}{\partial t} + v \nabla b = 0 \quad (2.8)$$

where v is the velocity vector and b is the brightness function. In addition, one can directly determine one-dimensional velocity in one-dimensional case, which is provided that the spatial derivative does not vanish. Also, the brightness is continuous. In the scope of aperture problems and scenes under varying illumination conditions, given 2D motion and apparent motion are not equivalent [17, 77, 100]. The OF methods are used for generating dense flow fields by computing the flow vector of each pixel under the brightness constancy constraint [30], its computation is detailed in the literature [101–103]. Extending OF methods to compute the translation of a rectangular region is trivial. Shi and Tomasi [104] proposed the KLT (Kanade-Lucas-Tomasi) feature tracker. In ‘*Good Features to Track*’ study of Shi and Tomasi [104], KLT tracker’s basics are firstly presented. The corner detection algorithm of Shi and Tomasi is used in other image and video processing studies as well. In survey study of Yilmaz et al. [30], they talked about a translation equation which is similar to construction of the OF method proposed by Lucas and Kanade [102]. According to Shi and Tomasi [104], regardless of the method used for tracking, not all parts of an image contain complete motion information. This is known as aperture problem. For better understanding, for instance, only the vertical component of motion can be determined for a horizontal image intensity edge. The researchers have proposed to track corners to overcome this difficulty, or windows with a high spatial frequency content, or regions [104]. Note that, the corner point is a conjunction of two crossed line in given image, which is the essential part of the image intensity edge. In object D&T process, the image intensity edge(s) is generally used to determine the boundaries of the object(s) that this determination aims to segment the image as region-level or edge-level. In the Fig. 2.4, a schematic for a general object D&T system based on an OF tracker.

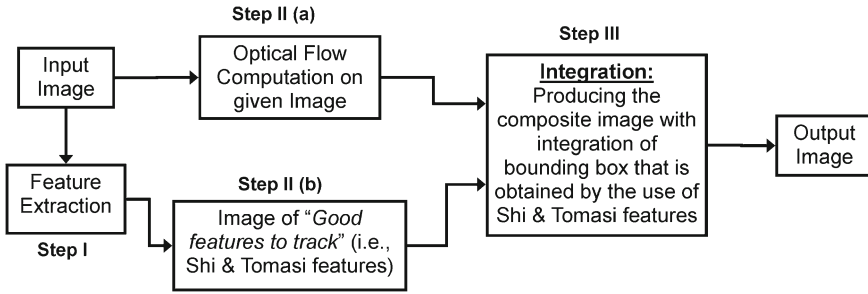


Fig. 2.4 The schematic for a general object D&T system based on an optical flow tracker

In Fig. 2.4, the Shi and Tomasi-based OF tracker takes an input image (or a frame of given video) at Step I, and then, the salient and significant features are extracted from given image. In Step II, there are two substep for OF computation and tracking. In Step II(a), the OF is computed on given current image (e.g., subsequent frame of given video, or an image other than first image), which is obtained from the flow between previous image (i.e., anchor frame of video) and given current image (i.e., target frame of video). This computation aims to show the direction of movement of moving parts based on the image velocity field. Also, in D&T process at Step II(b), the ‘*Good features to track*’ image is constructed only by using selected features which are optimally chosen due to Shi and Tomasi technique. Generally, the selected features indicates the OF-based corner points of tracked object(s). In Step III, the bounding box (or boxes) of target object(s) is integrated with given image (or a frame of given video) to construct the composite image. This bounding box (or boxes) is obtained by using the set of Shi and Tomasi feature points which indicated the moving object(s) in sequence of successive images (or video frames). All feature points are extracted in Step I, and then, some of them are defined as ‘*Good features*’ (i.e., optimally selected features) for given image in Step II (b). At the end, all integrations on the frame are shown on the output image, which involves bounding box (or boxes) for points of good features (i.e., corner points) for tracked object(s).

In the literature, the OF is a pixel-level representation model for motion patterns, in which each point in the image is assigned a motion vector, and, it is also known as a motion vector field [105]. Moreover, the techniques for estimating OF can be classified into three major categories, which are the phase correlation, block-based methods, and gradient-based estimation, respectively. Zheng and Xue [105] described these three categories in their study. According to their study, the phase correlation is a fast frequency-domain approach to estimating the relative movement between two images. Therefore, the block-based methods minimize the sum of squared differences or the sum of absolute differences, or maximize the normalized cross-correlation. It has been frequently used in video compression standards. Moreover, the gradient-based estimation is a function of the partial derivatives of the image signal and/or the desired flow field and higher order partial derivatives. In addition, the well-known

OF methods are belong to the aforementioned third category, which are the Lucas–Kanade (LK) method [102], Horn–Schunck (HS) method [101], Black–Jepson (BJ) method [106], and they are frequently used in the literature. By the way, we do not select the Black–Jepson method to implement in our study.

In this frame, we can consider in all aspects that there are two different main techniques for OF: Sparse OF and Dense OF. The most popular sparse OF tracking technique is the Lucas–Kanade OF, which is treated as locally constant motion. This method also has an implementation on OpenCV that it works with image pyramids, and allowing us to track the motions of faster objects [17]. This technique is referred as *Sparse OF* in our study. Another popular OF technique is a dense technique: the Horn–Schunck OF technique, which is treated as globally constant motion. This technique is referred as *Dense OF* in our study. Consequently, both kinds of implementations of OF tracker are used in our ViCamPEv software.

2.4.1 Horn–Schunck Technique (*Dense OF*)

The gradient-based methods are primarily based on the image flow constraint equation, which is derived from the brightness constancy assumption as well as the first-order Taylor series approximation [101, 107]. When one used only the image flow constraint equation alone, it is insufficient to compute the OF, because each equation involves two different variables. Also, a regularization approach is introduced by Horn and Schunck [101] that it employed a first order smoothness assumption to constrain the flow field and solve the flow [107]. According to the study of Teng et al. [108], the unsatisfactory performance of Horn and Schunck’s regularization method is mainly due to the insufficient number of iterations of their numerical method in the experiments. In addition, HS method can still generate very accurate OF as long as sufficient iterations (i.e., several thousands) are applied. However, the gradient-based methods suffer from some problems, such as illumination variations, image motion in vicinity of motion discontinuities (i.e., due to smoothness assumption), image aliasing, and noise [108]. According to the assumptions of the first-order Taylor approximation and the intensity conservation, the OF’s basic equation is as follows [109]:

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \quad (2.9)$$

This Eq. (2.9) can be transformed to another form shown in Eq. (2.10):

$$I_x u + I_y v + I_t = 0 \quad (2.10)$$

Yang et al. [109] also declared that, $I(x, y, t)$ represents the continuous space-time intensity distributing function of a given image. I_x and I_y , are the x gradient and y gradient of image I , respectively. In addition, I_t represents partial differentiation towards the time, and (u, v) indicates the components of the image velocity field.

In order to end up with dense flow estimates one may embed the OF constraint into a regularization framework. In the Horn and Schunck's study [101], they declared that the OF problem involves minimizing a combination of the OF constraint and a smoothness term [107]. They proposed the regularization method that it involves minimizing an energy functional of the following form [108, 110] :

$$E(u, v) = \int_{\Omega} (I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) dx \quad (2.11)$$

In Eq. (2.11), I is the image intensity function, $[u(x, t), v(x, t)]^T$ is the motion vector to be estimated, subscripts x , y , and t denote the direction in the partial derivatives, $x = [x, y]^T$ is a point in the spatial domain, Ω represents the 2D image domain and λ is a parameter controlling the degree of smoothness in the flow field [108]. Also, smoothness weight $\lambda > 0$ serves as regularization parameter: Larger values for λ result in a stronger penalization of large flow gradients and lead to smoother flow fields [110]. In dense flow fields, this regularization is a clear advantage over local methods. In addition, the gradient-based methods are sensitive to the image noise inherited from gradient measurement. The reader may refer for more detailed information about HS method to the studies of Teng et al. [108] and Bruhn et al. [110], respectively.

2.4.2 Lucas–Kanade Technique (Sparse OF)

There is a problem with the subdimensionality of the OF formulation in Eq. (2.11). Therefore, Lucas and Kanade [102] proposed another method to solve the problem on small local windows where the motion vector is assumed to be constant [111]. The LK method involves minimizing the equation of the following form:

$$\sum Z^2(x) [E_x u + E_y v + E_t]^2. \quad (2.12)$$

In Eq. (2.12), E is the pixel intensity, Z is a weighted window, and u and v are the motion vectors toward x and y directions [111]. When we look at details so that the goal of LK method is to align a template image $T(\mathbf{x})$ to an input image $I(\mathbf{x})$. Also, $\mathbf{x} = (x, y)^T$ is a column vector containing the pixel coordinates. According to the study of Baker and Matthews [112], if the LK algorithm is being used to compute OF or to track an image patch from time $t = 1$ to time $t = 2$ the template $T(\mathbf{x})$ is an extracted subregion (e.g., a 5×5 window) of the image at $t = 1$ and $I(\mathbf{x})$ is the image at $t = 2$. Baker and Matthews discussed about the parameterized set of allowed warps, which is denoted as $\mathbf{W}(\mathbf{x}; \mathbf{p})$. Also, $\mathbf{p} = (p_1, \dots, p_n)^T$ is a parameter vector. The warp $\mathbf{W}(\mathbf{x}; \mathbf{p})$ takes the pixel \mathbf{x} in the coordinates frame of the template T . Therefore, the warp maps it to the subpixel location $\mathbf{W}(\mathbf{x}; \mathbf{p})$ in the coordinate frame of the image I . As we mentioned before, the LK algorithm aims to minimize the sum of squared error between two images, the template T and the image I warped

back onto the coordinate frame of the template [112]. The mathematical expression is given as Eq. (2.13):

$$\sum_{\mathbf{x}} = [I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x})]^2. \quad (2.13)$$

In addition, warping I back to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$ requires interpolating the image I at the subpixel locations $\mathbf{W}(\mathbf{x}; \mathbf{p})$. The minimization of the expression in Eq. (2.13) is performed with respect to \mathbf{p} and the sum is performed over all of the pixels \mathbf{x} in the template image $T(\mathbf{x})$. The reader may refer for more detailed information about LK method to the studies of Sahba et al. [111], and Baker and Matthews [112].

According to the study of Yin et al. [113], in order to reduce computational cost, image corner detection is frequently used in OF-based trackers. Also, the pyramid implementation is often used in the literature (e.g., the OpenCV's implementation is simply based on Gaussian pyramidal decomposition of the image, which calculates coordinates of the feature points on the current video frame given their coordinates on the previous frame. The OpenCV's function finds the coordinates with subpixel accuracy [77]). When one looks at the details for corner detection, the extraction of the feature points is the key to effectively track the moving object(s). Therefore, the corner detection can be used to track target object(s). In the literature, the frequently used corner detection schemes are KLT detection [104, 113] and Harris corner detection [22, 113]. The Harris scheme is usually sensitive to the noise. In general, suppose image A is a given image for KLT computation, the second order moment matrix G can be constructed by KLT algorithm as following:

$$G = \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.14)$$

In Eq. (2.14), ω_x and ω_y is the parameter of window size. The size of the window is $[(2\omega_x + 1) (2\omega_y + 1)]$, and (x, y) is the pixel location. Also, I_x and I_y can be computed as given below:

$$\begin{aligned} I_x &= \frac{A(x+1,y)-A(x-1,y)}{2} \\ I_y &= \frac{A(x,y+1)-A(x,y-1)}{2} \end{aligned} \quad (2.15)$$

The reader may refer to the studies of Bradski and Kaehler [77] and Yin et al. [113] for more details of the corner detection (i.e., KLT tracker) and LK OF algorithm based on the pyramidal implementation.

2.4.3 Literature Review

In the literature, the OF method is frequently used in image motion analysis. The computation of OF from an image sequence provides very important information for

motion analysis. This issue involves moving object D&T, moving object segmentation, and motion recognition [107]. In the study of Lai [107], a new motion estimation algorithm was presented that it provides accurate OF computation under nonuniform brightness variations. The proposed algorithm is based on a regularization formulation that minimizes a combination of a modified data constraint energy and a smoothness measure all over the image domain. In the study, the data constraint was derived from the conservation of the Laplacian of Gaussian (LoG) filtered image function, which alleviates the problem with the traditional brightness constancy assumption under nonuniform illumination variations. Also, the resulting energy minimization was accomplished by an incomplete Cholesky preconditioned conjugate gradient algorithm. According to Lai, the comparisons of experimental results on benchmarking image sequences by using the proposed algorithm and some of the best existing methods were given to its superior performance [107].

Teng et al. [108] presented a very accurate algorithm for computing OF with nonuniform brightness variations that the proposed algorithm is based on a generalized dynamic image model in conjunction with a regularization framework to cope with the problem of nonuniform brightness variations. Also, they employed a reweighted least-squares method to suppress unreliable flow constraints to alleviate flow constraint errors due to image aliasing and noise, thus leading to robust estimation of OF. In addition, they proposed a dynamic smoothness adjustment scheme, and also employed a constraint refinement scheme to reduce the approximation errors in the first-order differential flow equation. According to Teng et al. [108], the experimental results show that their proposed algorithm compares most favorably to the existing techniques reported in the literature in terms of accuracy in OF computation with 100% density.

Myocardial motion is directly related to cardiac (i.e., heart) vascular supply. Also, it can be helpful in diagnosing the heart abnormalities. In addition, the most comprehensive and available imaging study of the cardiac function is B-Mode echocardiography [111]. According to the study of Sahba et al. [111], most of the previous motion estimation methods suffer from shear, rotation, and wide range of motions due to the complexity of the myocardial motion in B-Mode images. They introduced a hybrid method based on a new algorithm that it called combined local global OF which is in combination with multiresolution spatiotemporal spline moments. Also, it is used in order to increase the accuracy and robustness to shear, rotation, and wide range of motions. In their study, the experimental results demonstrated a better efficiency with respect to other B-Mode echocardiography motion estimation techniques such as LK, HS and spatiotemporal affine technique.

In the study of Yin et al. [113], in order to realize the detection of the mobile object with camouflage color, a scheme based on OF model was put forward. At first, the OF model was used to model the motion pattern of the object and the background. Then, the magnitude and the location of the OF were used to cluster the motion pattern, and the object detection result was obtained. At the end, the location and scale of the object were used as the state variables. Also, the Kalman filter was used to improve the performance of the detection, and the final detection result was

obtained. According to the study of Yin et al. [113], the experimental results show that the algorithm can solve the mobile object detection satisfactorily.

According to the study of Madjidi and Negahdaripour [114], despite high variability in the conditions of various bodies of water, a simplified image model allows the researchers to draw general conclusions on the computation of visual motion from color channels, based on average common medium characteristics. The model offers insight into information encoded in various color channels, advantages in the use of a certain color representation over others, consistency between conclusions from the theoretical study, and from experiments with data sets recorded in various types of ocean waters and locations. Their study concludes that OF computation based on the *HSV* representation typically provides more improved localization and motion estimation precision relative to other color presentations. In their study, results of various experiments with underwater data were given to assess the accuracy [114].

Amiaz et al. [115] presented a readily applicable way to go beyond the accuracy limits of current OF estimators. According to Amiaz et al., modern OF algorithms employ the coarse to fine approach. Also, they suggested to upgrade this class of algorithms, by adding over-fine interpolated levels to the pyramid. In addition, theoretical analysis of the coarse to over-fine approach explains its advantages in handling flow-field discontinuities and simulations show its benefit for subpixel motion. They reduced the estimation error by 10–30% on the common test sequences by applying the suggested technique to various multiscale OF algorithms. Using the coarse to over-fine technique, they obtained OF estimation results that are currently the best for benchmark sequences.

Kalmoun et al. [116] considered in their study the problem of 3D OF computation in real time. The 3D OF model was derived from a straightforward extension of the 2D HS model and discretized using standard finite differences. They compared the memory costs and convergence rates of four numerical schemes: Gauss–Seidel and multigrid with three different strategies of coarse grid operators discretization: direct coarsening, lumping, and Galerkin approaches. According to the study of Kalmoun et al. [116], the experimental results to compute 3D motion from cardiac C-arm computed tomography (CT) images demonstrated that their variational multigrid based on Galerkin discretization outperformed significantly the Gauss–Seidel method. In addition, the parallel implementation of the proposed scheme using domain partitioning shows that the algorithm scales well up to 32 processors on a cluster [116].

Fernández-Caballero et al. [117] introduced a new approach to real-time human detection. The approach process the video captured by a thermal infrared camera mounted on the autonomous mobile platform. Their approach starts with a phase of static analysis for the detection of human candidates through some classical image processing techniques. These techniques are the image normalization and thresholding, etc. Then, their proposal starts a dynamic image analysis phase based in OF or image difference. In their study, LK OF was used when the robot is moving, while image difference is the preferred method when the mobile platform is still. The results of both phases were compared to enhance the human segmentation by infrared camera.



<http://www.springer.com/978-1-4614-6533-1>

Performance Evaluation Software
Moving Object Detection and Tracking in Videos
Karasulu, B.; Korukoglu, S.
2013, XV, 76 p. 11 illus., Softcover
ISBN: 978-1-4614-6533-1