Evolution of hand tracking algorithms to MirrorTrack

Yannick Verdié

Abstract—When recognized by a computer, hands can be very powerful and natural tools to convey information. Hand tracking techniques evolved a lot during the last years from one camera solutions to 3D solutions, offering users an improved experience. This paper first describes (1) the main methods used in hand tracking algorithms based on single camera and their limits, (2) the meaning of touch and the advantages and particularities of this newly rehabilitated sense and (3) how single camera algorithms have been improved by stereo analysis (3D). We concluded that Stereo Analysis is a very promising area which still need research efforts, especially the very convenient solution of MirrorTrack.

Index Terms—Image Processing, Hand Tracking, Single camera, Stereo analysis. Touch, MirrorTrack,

1 INTRODUCTION

Since centuries, human beings have been trying to create and improve tools. However, one of the most natural and effective tools that exist is the one we got from nature: our own hands. Our hands, and especially our fingers, are the most convenient tool to use. If we make possible the interaction between this natural tool and modern tools such as computers, it becomes a very powerful tool.

In order to allow the empowering of hands as tools by making them interact with a computer, the mediation of a camera or captors is necessary. For several years, researchers have been trying to implement the best solution possible to make this interaction work. This paper will highlight the main techniques aiming at detecting and tracking fingertips that have been published. Hand tracking techniques evolved a lot during the last years from one camera solution to 3D solutions, offering users an improved experience. Although the processes introduced in this paper could be used in many different goal, this paper will be focused on the definition of the fingertips location.

Indeed, the usage of hand as an input device substituting the computer mouse function is an attractive idea. The huge success of embedded touch devices such as the iPhone of Apple lead to think that this complex area of research still hide some geese laying golden eggs. However, even if a lot of research work has already been done, a completely satisfying solution has still to be found. MirrorTrack, as will be demonstrated at the end of this paper, takes advantage of existing algorithms while introducing particularities that make it a very powerful solution. It is a very promising and interesting area of research for the coming months.

This paper will present the evolution of research work on Hand tracking. The one-camera solution and its limits will be presented first, including the main algorithms used which will be of interest for the other solutions also. Then, the importance of touch will be explained. Finally, the last solutions using 3D will be described.

2 SINGLE CAMERA (2D LOCATION)

The main advantages of using a single camera is that users could easily use their own camera or webcam. It is also technically less complicated: it allows a single common domestic computer to handle the algorithm. It is very important to always think in term of efficiency in function of resource needs. There is always the risk to develop a very efficient algorithm that will be impracticable for the user, that is to say useless in the majority of cases. It is especially the case when complex algorithms result in too long computation. Researches [1] and [2] defined the frequency limit of the real time as 20Hz. Above this limit, the delay felt by the users is described as uncomfortable. Michotte reports in [3] that a user feels as an "immediate causality" two successive events which the delay between them is less than 50ms (f > 20Hz). Single camera was initially exploited to localize in a plane the location of the target. Methods were found to simulate and estimate the 3D location of the hand with just one camera. In this chapter, the more utilized algorithms aiming at localizing the position of fingertips will be presented. Most of the time, a common chain of process is used. The first common step is the foreground/background segmentation used to start drawing limitation between the target in front of the camera (hand, face, body, and so on ) and the background. The second step is fingertips location. Generally, the third step is the statistical estimation of fingertips location. This last step is optional, but the improvements brought by this step make it necessary. (see figure 1)

Not all those steps are mandatory and depending of the scientists’ needs, one algorithm or another will be used. After describing the research work developed for each of those 3 common steps, the limits of those solutions will be explained.
period initialized at the beginning of the process. A hand dataset of user’s hand pictures created during a learning process. Although it is true that scientists created new specific solutions according to their needs, some of them are recurrent and could be classified in those following sections:

2.1 Foreground/background segmentation

Foreground/background segmentation is an image processing operation aiming at differentiating the foreground from the background of a picture. There are many implementations of this process. Although it is true that scientists created new specific solutions according to their needs, some of them are recurrent and could be classified in those following sections:

2.1.1 Filter

- Polarized Filter The segmentation is proceeded directly by using a rotated polarizing filter on the camera. It creates the illusion of a screen switched off due to the polarization of the LCD’s light. Contrary to the LCD’s light, the light reflected by the hand is not polarized and is detected by the camera. [1] used this process to directly disable the content of the screen and used a threshold to obtain the hand’s shape. This process has the propriety to be very easy to implement and do not need a lot of computer’s resources but need usage of an external component. This need is quite restrictive because it is not easy to find a filter that works with the camera; It brings an external cost that all users are not willing to pay.

- IR filter An Infra-Red camera could also be used to segment foreground from background as in [4] or [5]. The skin is revealed easily. [6] use this segmentation as a pre-segmentation to improve the result of an other foreground/background segmentation by graph cuts algorithm. [7] and [8] combine a back-projected display with IR camera(s) to detect fingertips only when they touch the screen. Those high resolution techniques allow researchers to process directly the shape of the touch which makes possible the distinction between hovering and click. [9] proposes to setup on LCD back screen a network of IR captor (detector + emitter) that detects the proximity of fingertips. As for the polarized filter, adding an external component reduces the computer recourses usage but brings an external cost and is not easy to setup. Not all cameras provide IR system or support filter add-on.

2.1.2 Color segmentation

The goal of the color segmentation solution is to detect the hand based on the specific color of the skin. [10] uses a dataset of user’s hand pictures created during a learning period initialized at the beginning of the process. A hand segmentation with Bayes decision theory is then used to define for each pixel whether or not it is a skin pixel. [11] uses this same approach and exploits the dataset for a histogram-based skin classifier described in [12]. A lot of techniques and derive aim at classifying a pixel as a skin-pixel or a non-skin pixel based on a non parametric approach. The goal of this approach is to estimate whether or not the pixel is a skin-pixel ([13], [14] and [15]). The chosen color space for this process is also important. HSV color representation seem to be the better chose [16] because it avoids the influence of shadow and lighting variation by discarding the V channel. Moreover, while working in this chromatic color space (H,S), [17] shows that the distribution of skin-pixel samples of “253 persons from different ethnicities, e.g. Caucasian, Asian, African and dark-skin types” could be roughly drawn as a gaussian mixture model built with 4 gaussian bells. This propriety could be exploited to detect the skin-color without having to do a learning period. The result is doubtless improved.

Unfortunately, those processes based on skin-color have some drawbacks. The background could have some similitude in term of color resulting in a non-segmentation of the hand and the background. In other terms an algorithm based only on the color to detect an object will grab every objects having closely the same color. Another problem could be the influence of the external color light on the skin. For instance, with an incandescent lighting, the skin color will seem more yellow that it is in reality. Using the HSV space could avoid this kind of issues, but what if the lighting color is blue? In this case, this is not only the value (V) that changed but also the hue (H) and saturation (S).

2.1.3 Frame differencing

Frame differencing is an image processing operation aiming at detecting a new event in a already known situation. First of all, there is a period of “learning” where a model of comparison, also called reference, is defined. All the various methods used define a reference frame from where the current frame will be subtracted. [11], [18] or [19] use a simple technique by grabbing just one reference frame during the beginning of the algorithm based on the assumption that the user’s hand will not be in front of the camera at this moment. But if that happen, the hand will be stored in the reference and the efficiency of this method will be jeopardized. This simple process is still wildly used in case of static background.

Nonetheless, there is a huge drawback in case of dynamic background because the reference will not match the background anymore. A background becomes dynamic when the user open windows. [2] adapts this process to avoid its drawback by using a smart reference update runtime. The algorithm switch now from a stage of learning, to one stage of process on an iterative manner. [20] uses statistical approach by storing for each pixel 4 tuples, the expected color value, the standard deviation of color value, the variation of the brightness distortion, and the variation of the chromatic values. Once the reference is created, a frame is grabbed on which a subtraction from our reference frame is done in order to
highlight all moving objects: that means all objects that were not on the reference but on the current frame. In 2004-2005, Kyungnam kim, Thanarat H. Chalidabhongse, David Harwood and Larry Davis proposed in [21] a new algorithm composed of a learning period and a process period as shown previously but with some differences. This algorithms proposes to store not a reference per frame as shown previously but a reference per pixel, introduced as a "box" in their paper. The dynamic processes of learning period and process period are kept and every boxes are separately updated according to criteria. This algorithm is very efficient but use substantial resources. The main problem with the frame differencing is the difficulty to update smartly and frequently the background without including the foreground inside. [2] and [21] found a solution to this problem, but it is still not completely satisfying because each solution bring some other issues.

2.1.4 Projection

All the one-camera solutions have a common drawback which is occlusion. The hand can be hidden (by the other hand, the user body) and then not detected anymore. Instead of putting the camera in the same side than the user, the camera can be placed behind the screen. This solution tackles directly the occlusion problem at the root. [7] and [8] use this installation to detect the touch on the screen thanks to the fingertips' shadow and in same time avoid occlusion issue. This configuration seems to avoid all major issues inherent to the single camera configuration, but unfortunately, it is impossible to detect efficiently the finger hovering which is very interesting [22] and require complicated and inconvenient installation.

2.2 Fingertips location

The goal of fingertips location is to localize the fingertips of the hand. Several different techniques exist such as curvature based on the hand boundaries, region, based on the whole hand shape, classification, based on a lexical grammar, and markers, based on tracking reference of the hand.

2.2.1 Curvature

The curvature algorithms are based on the boundaries of the hand to localize the fingertips. The algorithms determine the presence of fingertips and their coordinates by analyzing the curve of the hand, its convexity, and so on. [23], [11] and [18] exploit the hand curve by creating two vectors from a list of points representing our hand’s boundaries. By this way, fingertips are localized easily by looking at the value of the angle between two vectors. Of course, there will be more than one point at the hand extremity candidate to be a fingertips. Each project used its own technique (central moment of those points, more probable candidate, and so on) to handle this problem. [24] used a mixed solution between curvature and region to found the fingertips by rotating all the shape, up to finding the longest projection of the shape. The curvature is then proceeded.

However, to be implemented, the curvature solution needs a perfect curve. It does not work properly in case of occlusion.

2.2.2 Region

The region algorithms are based on the filled form of the hand to localize the fingertips. The algorithms proceed usually by looking at the ratio of the area filled to a parametric area. [2] uses two diametrically different filled circles to find the location of a finger and the thumb. [25] uses a square of 7 x 7 pixels to localize fingertips. Virginia Tech HCI students used a star algorithm processing a binary black and white shape of the hand by looking at the number of star’s arms being in the white area and the number being in the black, for each pixel. They used a 8 arms star and defined the fingertips as being the group of pixel with one or less star’s arm in the white area. [26] uses the shape filtering on binary pictures. A very different approach also based on region could be to use snake algorithm to segment the hand. Snake aims at minimizing the external energy acting on an initial closed curve. At each iteration, the snake computes new positions minimizing its energy [27].

The limit of Region solution is that it lose efficiency by searching fingertips on every filled form pixels.

2.2.3 Classification

The classification algorithms aim at seeking in the frame a shape that closely match a shape library used in this algorithm, also called lexical grammar. In other words, according to some parameters that define when the similarity between a picture in the classification and the current frame means a positive response, it is possible to detect every hand’s shape in a picture. This process is intensively used to detect and localize faces in pictures but seems to have some drawbacks in some circumstances to localize hand. Nevertheless, [1] uses it to find fingertips and [10] creates its own hand picture dataset with 326 images divided in two categories: half for the training data, and half as test data. [23] uses a Bayesian classifier to segment the hand from the background.

2.2.4 Markers

[28] uses markers to easily define hand geometry and fingertips location. Markers are colored signs positioned smartly on the hand in order to allow the algorithm to track their location and to rebuild virtually the hand. This kind of process is commonly used to track movements of all the body, and is called motion tracking. Using this process for a hand led [28] and [29] to localize the hand in 3D thanks to just one camera and one 3D hand skeleton. By inverse kinetics, the missing information of the 3D localization and the gesture of the hand are estimated by "posture hypothesis validation".

2.3 Estimation

The last common step aiming at statistic estimations. An interesting overview of the main algorithms used to estimate location and trajectory is available in [30].
2.3.1 CAMSift

Based on the probability distribution of color target, this algorithm is commonly used to detect the central moment of the target. By tracking a hand, CAMSift will process not the estimate shape but the location of its central moment. [31] uses CAMSift to draw letter with his hand, the whole hand being used as a pen. Nevertheless, it is also possible to draw the hand shape. [32] did that by using an histogram back projection after using CAMSift algorithm. [33] uses a new version of histogram back projection to determine hand’s shape and hand’s central moment. The results observed are less convincing than those of the following algorithms.

2.3.2 Kalman

The Kalman algorithm is a recursive filter based on previous measurements. That is why the Kalman algorithm is composed of two distinct stages: Predict and Update. The predict stage processes the current measure (that includes the real position and the noise inherent to the measure process) with the model to obtain an a priori estimate. The update stage aims at including in the model new measurements to obtain an improved a posteriori estimate. Kalman algorithm estimates where should be the target based on the memorization of data characteristics of its previous known localization. A model maximizing the a posteriori probability of those previous measurements is build. This model takes into account that previous measures used include some incertitude as well as the current measure of the target’s localization. The filter considers its own incertitude and the new incertitude of the current measure to compute a location that corresponds to the most probable localization. Therefore, for each frame, the model is adjusted for the next estimation. [33] uses Kalman algorithm twice on each hand to estimate the locations of the hands’ central moments. [19] use it in their TAFFI (Thumb and Fore-Finger Interface) algorithm to smooth the trajectory of the virtual mouse.

The main problem is that Kalman is a parametric filter. As said in [34], "the Kalman filter models a single hypothesis [that] is unimodal Gaussian, it is not possible to represent multiple hypotheses simultaneously". Kalman algorithm works on one point only and it is become very complex to use it for the ten fingertips.

2.3.3 Particle filter (i.e. Condensation, MCMC-PF)

The particle filter is a kind of non-parametric algorithm: instead of estimating mean, variance, or other parameters, such as Kalman filter. Particle filter algorithms are based on a Bayesian technique. Moreover, Particle filter uses non-linear motion models which is more representative of the common day-to-day trajectories. A very famous algorithm is Condensation (Conditional Density Propagation), presented in [35]. However, it has some drawbacks tackled by a Markov Chain Monte Carlo (MCMC PF) introduced in [36] and [37].

A very relevant comparison between Kalman filter and Condensation was done in [38]. I shows that in order to track a missile (single point), Kalman give better results but take more time to converge than Condensation.

2.4 Issues

This paper has described the different algorithms used in the 3 common steps involved during hand tracking with a single camera. All have in common the limitation inherent of one-camera solutions which are the occlusion and the leak of touch analysis.

2.4.1 Occlusion

A single camera could be occluded easily. The user movement around the working space is restricted because he cannot put his/her head or his/her body between the hand and the camera field of vision. There are no real alternative to avoid occlusion, that is why many researchers choose to use more than one camera to always keep a free angle of view.

2.4.2 Touch detection unavailable

More importantly, the touch detection is unavailable with one camera solutions. In fact, it is almost impossible to define the 3D hand location with a single camera. This issue results in a slow evolving popularity of hand tracking because users fill uncomfortable to use a mouse substitution without the possibility to be sensitively aware of a click. There are many studies on the aesthetics of touch revealing that touch has been neglected for a long time because of the difficulty to share the experience of this sense [39].

Researchers try to create some touch sense ’s substitution using single one camera.

Techniques of touch simulation:

* Delay detection (i.e. Staying n second in the same location means a touch) [25] tracks fingertips and defines a click when the fingertips stand 2 seconds on the same location.
* Gesture detection (i.e. Using thumb and index as a gun trigger) [40] tracks the gesture of the hand. Four classes of gestures are defined such as "Point, Reach, Click, and Ground". For example, a Click is detected when the thumb and index in gun form are "triggered" together.
* Color nail modification detection [41] uses the modification of nail color when the fingertips push on the surface to detect a click.

To conclude, even if some techniques have been implemented, the detection of touch is not successfully available with a single camera. The simulation techniques all have limits because there are not convenient, natural or reliable. However, the sense of touch is primordial in the creation of a better and
pleasant interface between the human and the computer. The touch reveals itself to be very important as described in the following section.

3 MEANING OF TOUCH

The sense of touch has been neglected for a long time in favor of the senses of sight and hearing. The first reason is that it seems more important for the user to be surrounded by pictures and sounds than feeling through touch. It is also more difficult to communicate information interactively through touch in opposition to sight and hearing. But has shown in [39], all the reasons against the usage of touch seem to not be valuable anymore. People are used to be surrounded by sounds and pictures and become accustomed to this exiting wave of sound and light. The touch proves itself to be relatively new and unexploited in HCI. Communication through touch becomes very popular and demanded these last years.

Mouses and keyboards tend to be progressively replaced by single touch screens commonly used in PDA or ATM because there are "direct interaction system[s]"[42]. Dr Quek\textsuperscript{1} defines Touch as an "immediate effective media of expression". A lot of information are communicated with a simple touch. It is "a visceral form of human interaction". Moreover, [43] defines Touch as unique among the five senses, because it is the only one that gathers information on the "innards" of the object, which means not only information about the surface, but also in depth information such as texture, temperature, and so on...

As [44] concluded in their early study, "the combination of tactile output with a pointing device such as the mouse is quite natural". In a nutshell, Touch has a huge potential of interaction still unexplored. It could be a channel of multi-modal communication leading to a better immersion of the user in the real world; the bridge between them being more thin and transparent.

The need of improved solutions to provide a better experience to the user leads to a new approach: The Stereo analysis. 3D location allowed by Stereo analysis could manage the touch detection, and also handle the occlusion problem introduced in the previous section.

4 STEREO ANALYSIS (3D LOCATION)

Stereo analysis was used in various applications from hand and body tracking to meteorological satellite data process [45]. Stereo analysis uses more than one camera. To overcome the location issue inherent to single camera usage, as well as to implement a robust system using touch sense, Stereo analysis takes advantage of multi field of vision created by several cameras. Stereo analysis uses the same algorithms that have been described in the first part of this paper but add the several cameras strength.

Stereo analysis setup requires a mandatory first stage of calibration for each camera separately. It is followed by a stage of computation aiming at defining parameters to achieve 3D reconstruction of the hand position and hand contours. A new specific issue emerges with Stereo analysis which was unknown during the single camera process. All the 2D data from each camera have to be merged into a reliable 3D information. The step of calibration and the mechanism of merging data together are explained in [24]. This work uses two basic cameras and despite the usage of two particle filters to estimate fingertips location, the algorithm proceeds frame in real time.

Of course, because of the noise and possible calibration errors, the shape from each camera could not exactly converged into one single 3D shape. In order to however compute the more probable shape and location of the target, a common estimation called Least-Square estimation is proceeded [45]. As said in [18], "by using stereo vision we can not only determine contact information, but also the distance of a fingertips from [ the] surface for additional types of input".

The project described in papers [15] and [23] uses two simple cameras. In those papers, the problem of matching the two 2D shapes together is defined to work like the stable marriage problem. The stable marriage problem aims at successfully creating an optimal matching from two entities. The optimum is met when all couples are created and there is no possible combinations of two elements that is better than the existing one.

[46] does a "review on vision-based full DOF hand motion estimation” where gesture recognition is introduced as an old issue never completely solved. A full reconstruction of the hand shape was previously too complex, and projects usually focus on the extraction of the desired information(location of fingertips, orientation of the hand, and so on). Nowadays, an alternative solution to the full reconstruction was found by using either a kinetic structure alone or a kinetic structure mixed with reverse kinetic functions. A kinetic structure is a skeleton of the target including structure restriction (as length and/or rotation). It is defined by its number of Degree Of Freedom (DOF), that means the number of rotation and translation allowed by the skeleton. It is useful to improve tracking object once the skeleton was attached to the target as well as gesture recognition.

An other solution that seems the most promising is MirrorTrack. Keeping in mind the need of touch detection and the problem of occlusion, MirrorTrack [47] provides a cheap and efficient system using two cameras to detect hovering, touch, and hand 3D location. Moreover, another important strength of this project is its adaptability. Two common webcams could be setup on any common screen, TV, LCD, are the only necessary required devices to make the project work. Taking advantage of the mirror effect of screens looked in a low-azimuth angle, MirrorTrack detects easily the distance between the fingertips and the surface by looking at the finger tip’s reflection. It also handles the problem of occlusion by installing cameras close to the surface.

1. HCI director and professor in CS department, Virginia Tech, VA
5 Conclusion

Several-camera solutions provide an improved user’s experience because they take advantage of a rehabilitated sense, Touch. Those solutions are promising but the research in that area is not very developed yet.

MirrorTrack especially seems to be an interesting solution to analyze as it is both very easy to implement and providing a good user’s experience by using a completely new method. Its main particularity is that it is inexpensive and can be setup with simple day-to-day devices. It opens the door to an enriched Touch experience for numerous users. Hands are ready to become powerful tools for the greater number.

### Table 1: Synthesis of the algorithms used. Some of the projects are dedicated to hand tracking, others are focused on more theoretical topics.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>camera</th>
<th>F/B segmentation</th>
<th>Finger location</th>
<th>Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>1</td>
<td>Polarized filter</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[6]</td>
<td>2</td>
<td>InfraRed filter</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[4]</td>
<td>1</td>
<td>InfraRed filter</td>
<td>Classification</td>
<td>Kalman</td>
</tr>
<tr>
<td>[5]</td>
<td>1</td>
<td>InfraRed filter</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[7]</td>
<td>2</td>
<td>InfraRed filter</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[8]</td>
<td>1</td>
<td>InfraRed filter</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[9]</td>
<td>1</td>
<td>InfraRed filter</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[10]</td>
<td>1</td>
<td>Color segmentation</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[13]</td>
<td>1</td>
<td>Color segmentation</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[14]</td>
<td>1</td>
<td>Color segmentation</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[15]</td>
<td>2</td>
<td>Color segmentation</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[16]</td>
<td>1</td>
<td>Color segmentation</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[12]</td>
<td>1</td>
<td>Color segmentation</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[14]</td>
<td>1</td>
<td>Color segmentation</td>
<td>Classification</td>
<td>None</td>
</tr>
<tr>
<td>[11]</td>
<td>2</td>
<td>Frame differencing + Color segmentation</td>
<td>Curvature</td>
<td>None</td>
</tr>
<tr>
<td>[18]</td>
<td>1</td>
<td>Frame differencing</td>
<td>Curvature</td>
<td>None</td>
</tr>
<tr>
<td>[19]</td>
<td>1</td>
<td>Frame differencing</td>
<td>Curvature</td>
<td>Kalman</td>
</tr>
<tr>
<td>[2]</td>
<td>1</td>
<td>Frame differencing</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[20]</td>
<td>1</td>
<td>Frame differencing</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[21]</td>
<td>1</td>
<td>Frame differencing</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[7]</td>
<td>1</td>
<td>Projection</td>
<td>Curvature</td>
<td>None</td>
</tr>
<tr>
<td>[8]</td>
<td>1</td>
<td>Projection</td>
<td>Curvature</td>
<td>None</td>
</tr>
<tr>
<td>[22]</td>
<td>1</td>
<td>Projection</td>
<td>Curvature</td>
<td>None</td>
</tr>
<tr>
<td>[7]</td>
<td>1</td>
<td>Projection</td>
<td>Curvature</td>
<td>None</td>
</tr>
<tr>
<td>[23]</td>
<td>2</td>
<td>Classification</td>
<td>Curvature</td>
<td>None</td>
</tr>
<tr>
<td>[24]</td>
<td>1</td>
<td>Classification</td>
<td>Curvature</td>
<td>None</td>
</tr>
<tr>
<td>[25]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[26]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[27]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[28]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[29]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[31]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[32]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[33]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[36]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td>[37]</td>
<td>1</td>
<td>Classification</td>
<td>Region</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CAMSift with back reconstruction</td>
<td>CAMSift</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CAMSift</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Particle filter</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Particle filter</td>
<td>None</td>
</tr>
</tbody>
</table>
References


