A Review of Object Tracking Techniques and Processes for 4k Video Lecture Editing

Maximilian Hahn
University of Cape Town
Cape Town
South Africa
hhnmax001@myuct.ac.za

ABSTRACT
In this paper we review multiple object tracking algorithms and the subcomponents of object tracking. This is for the purpose of selecting an algorithm to implement into a 4k lecture video editor that will output a 720p resolution frame that follows the lecturer. The solution needs to be accurate enough to track the lecturer in a reliable manner and efficient enough to process in less than a day. In this review we discuss object representations, feature analysis, object detection, object tracking, handling occlusion and processing hand gestures. We concluded that many algorithms exist that tailor to efficiency or reliability but only few are able to both while being applicable to our problem. We found that occlusion can be handled by recording velocity and orientation data of objects. Processing of hand gestures is made easier with tight fitting representations such as silhouette and contour. Both kernel tracking and silhouette tracking show the most promise for our needs.

Keywords
Object Tracking; Lecturer Tracking; Review

1. INTRODUCTION
The field of computer vision is centered on the analysis, understanding and reproduction of human vision in a programmatic system [Sonka et al. 2008]. Applications of this field includes facial recognition, instance recognition, category recognition as well as context and scene understanding. This is done by utilizing many image formatting, image processing, segmentation and feature-based approaches [Szeliski 2010].

The purpose of this review is to discuss algorithms for tracking a lecturer in a lecturing environment. This tracking needs to include hand gesture recognition and occlusion handling. References to implementations of the algorithms will also be discussed. The purpose of this is to have an understanding of tracking algorithms in order to choose an approach for the 4k video tracking project we plan on implementing. This project will have one stationary 4k video camera and the algorithm is projected to cut out a 720p slice of the video that will virtually pan around the 4k environment tracking the lecturer.

The processes behind object tracking can be simplified to object representation, feature selection, object detection and object tracking [Yilmaz et al. 2006]. Object representation is the chosen format for any element in the scene of interest that is then noted for further analysis. Feature analysis takes a set of candidate features and selects from those a subset that best fit the criteria required, thus reducing the number of features required for further processing [Jain et al. 1997]. Object detection utilizes the subset found in feature analysis to define objects in a certain criteria [Yilmaz et al. 2006]. Lastly object tracking makes use of the objects found in the previous step to track the progression of an object across the scene [Yilmax et al. 2006]. An example specific to our goal is face and body tracking as in Lerdsudwichai et al. (2004) in this paper a surveillance system is described that can track multiple people and handle all forms of short-term occlusion. Handling partial or total occlusion is a problem that some object tracking algorithms do not manage as described in Nguyen et al. (2004). Gestures are an important additive to understanding a speaker [Mitra et al. 2007] and come in many forms each needing to be differentiated by the algorithm implemented. Wang et al. (2004) describes a system of gesture tracking used for lecture video editing chosen because it is critical for the students that the speaker is best understood.

2. OBJECT REPRESENTATION
Objects representation schemes are used as markers for tracking objects and are chosen based on the context of the scene and the tracking algorithm used. Yilmaz et al. (2004) uses a contour based approach to represent objects that have highly variable contours such as people. This also provides detailed edge features as opposed to using a primitive representation. For example, where the edges are specified simply. Objects can be represented as a point or a cluster of points, primitives, silhouettes, contours or as a skeleton [Yilmaz et al. 2006].

2.1 Point Representation
Objects can be represented by a point central to the object as in Veenman et al. (2001) or using a set of interest points per object as in Serby et al. (2004). Objects as a central point require the tracking of a static number of points [Veenman et al. 2001]. This tracking is made difficult by the motion correspondence problem described in Rangarajan et al. (1991) where points in one frame need to be paired up with points in another frame given that the points can move. Putting constraints on the input helps to minimize the motion correspondence problem as in Veenman et al. (2001) in this case the number of points is fixed for all frames allowing their *Greedy Optimal Assignment* algorithm to track all points and detect occlusion. Rangarajan et al. (1991) uses a different approach assuming that objects in the world tend to move smoothly, paired with an optical flow algorithm. The algorithm in (Rangarajan et al. 1991) does not handle full occlusion as Veenman et al. (2001) did. Veenman et al. (2001) offers a much better approach to our problem as we expect a lecturer to be partially occluded multiple times in a lecture and fully occluded once every few lectures. An important quality of interest points is their immunity to changes in both camera position and scene illumination (given that there is enough light to see the scene) [Yilmaz et al. 2006]. It is important that if light suddenly floods the lecture theatre objects aren't lost. The overall importance of this rests on the decision to do light equalization preprocessing for this project.

2.2 Primitives

This representation makes use of primitive shapes such as rectangles, circles and ellipses to specify the boundaries of the tracked object [Yilmaz et al. 2006]. These shapes are then translated around the scene and transformed to follow and fit the object they surround. Primitives are best suited to model rigid objects but can be used to model non-rigid objects especially if the primitives can be transformed. Comaniciu et al. (2003) proposes a kernel-based tracking technique that uses primitive representation that handles partial and total occlusion. (Gavrilla 1998) implements a multi-feature hierarchy of primitive templates but doesn't completely handle occlusion it only mitigates occlusion. The ability to check many types of primitives against an object isn't applicable to our project as we need only track people and could therefore decide on a single primitive beforehand. Both papers use primitive representations but Comaniciu et al. (2003) is more applicable to our project as we need to handle occlusion but primitives don't easily provide us with the gesture information we need especially given that people are non-rigid objects.

2.3 Silhouette and Contour

Silhouettes as object representations are generated using background subtraction where movement of the silhouette is compared per frame to track the object [Yilmaz et al. 2004]. The paper focuses around an implementation of contour-based object tracking. This is done by tracking active contours in order to generate a spline around an object. This implementation works well for cases of near-total occlusion and is definitely a candidate for our project. Parks et al. (2014) found that the knowledge that people are the subject being tracked can be used as an assumption to reduce the number of incorrect detections. This can also be used to reason through person-to-person and object-to-person occlusions by understanding typical human motion. Using a spatial awareness algorithm this implementation can predict entrances and exits of human objects as well as spot occlusion and re-track an object when possible. This implementation works especially well for our project as our camera is static so spatial calculations need only be calculated once and it handles occlusions well.

2.4 Skeletal Models

This model generates and tracks the skeleton of an object in a scene [Yilmaz et al. 2006]. Han et al. (2013) describes the *Microsoft Kinect's* skeletal tracking that uses a RGB camera with an attached depth sensor. While this implementation is effective the *Microsoft Kinect* is limited to a small room. For applications to a larger room a stronger depth sensor would be required. The 4k cameras used in our project don't have depth sensors so this isn't a viable solution. Skeletal models are useful for tracking and understanding sequences of human actions as in [Ali et al. 2001]. Actions such as sitting down and reaching for objects are geometrically represented easier by using a skeletal model. These actions can be used for in depth gesture analysis. Our project doesn't need that kind of depth for gestures so skeletal models aren't applicable for this project.

3. FEATURE ANALYSIS

A feature is some form of pattern in a frame. The selection of a feature to search for should be made on how unique the feature is [Yilmaz et al. 2006]. Feature analysis is used as preprocessing to object detection in order to make object detection quicker and more accurate. Features that are easier to distinguish will be found more consistently than others and will not be confused for other features. Features to search for are usually chosen manually depending on the application of the program. Common search features include: color, edges and optical flow.

3.1 Color

Feature selection can be done based on the color of a feature as in Phung et al. (2005). Skin color is a useful feature to search for when tracking people. The subset of skin colors that need to be checked are also limited to varying shades of brown and peach. An extra element of complexity is that lighting conditions play an important role in how the color of skin appears, this is a problem for all kinds of color features. Paschos (2001) concludes that the choice of color space has a noticeable impact on the performance of some color texture analysis algorithms. Specifically he concluded that HVS performed better than both RGB and L*a*b*. Phung et al. (2005) contradicts this by using four different color spaces (RGB, HSV, YCbCr and CIE-Lab) to achieve permissibly similar detection rates across the color spaces also using a segmentation algorithm. This contradiction likely stems from the form of testing that each paper did. As a precaution our project will make use of HVS.

3.2 Edges

Edge detection picks up boundaries of objects generally characterized by large changes in pixel color and luminosity [Maini et al. 2009]. An edge detector processing a noisy image can result in a distorted output edge. Preprocessing is often used to fill noise as in Weiss (2009) using median filtering and Ma (2000) using Gaussian smoothing. Between the two, median filtering performs better simply because precision is lost slower than with Gaussian smoothing. Canny edge detection [Canny 1986] is a highly popular and effective edge detection algorithm [Yilmaz et al. 2006] that employs a number of processing steps to reliably find edges in an image drastically reducing the amount of processing required to find objects. Applied to our project edges are very useful as it is difficult to completely occlude every edge of a person especially given that the cameras aren't stationed at head height.

3.3 Optical Flow

Optical flow describes a pattern of perceived motion of objects in a sequence of images [Horn 1980]. Discontinuities in the optical flow are a strong indicator of one or more objects in the sequence. Beauchemin et al. (1995) states that changes in brightness have a strong negative impact on optical flow algorithms. Even though strong contrasts are often a good indicator of features [Yilmaz et al. 2006] an object being flooded with light or dark between frames can confuse many feature analysis techniques. A common approach is Beauchemin et al. (1995) which found that occlusion in a sequence of images means no optical flow data can be generated for fully occluded objects, but that contour velocity calculations can be used to potentially track the occluded object. A preprocessing technique such as histogram equalization can be used to minimize changes in brightness between frames [King 1996]. García et al. (2013) states that optical flow suffers from long computation times. This is a problem for this project as our processing needs to be complete less than 24 hours after the lecture was recorded and ideally within 6 hours. Given that we are processing 4k video a computationally inefficient technique is not viable.

4. OBJECT DETECTION

Object detection runs in every frame or over a sequence of frames [Yilmaz et al. 2006]. Using a sequence of frames decreases the count of false detections by asserting that the object is consistently

detected across those frames. Objects are made up of one or many features and can be represented in many ways. Common approaches include point detectors which use point representations, background subtraction which creates silhouettes and contours and lastly segmentation which creates contours.

4.1 Point Detectors

Point detectors search a frame for interest points [Yilmaz et al. 2006]. These points are found by searching for a desired texture feature. The interest points representation is desirable because of its immunity to lighting and position changes. Mikolajczyk et al. (2005) compares 10 different interest region detectors working with image frames put under conditions of positional and rotational transformation. Final results recorded are recall, precision and number of correct matches. Gradient Location and Orientation Histogram (GLOH) performed best followed by Scale-invariant feature transform (SIFT). Memarzadeh et al. (2013) implements a GLOH algorithm for 2D detection of workers and construction equipment. This is similar to our project in that it tracks people. They reported that although the algorithm had a high rate of accuracy it suffered from execution time noticeably slower than real-time. They do go on to say that with the implementation of an NVIDIA CUDA framework they believe the algorithm would run in real-time however this is especially relevant to our project as while we don't need to run in real-time, if the algorithm turns out to run too slowly CUDA parallelization is a viable option. Yilmaz et al. (2006) confirms that SIFT performs well for recall, precision and number of correct matches but also that the algorithm runs quickly making it a candidate for our project as well. Lindeberg et al. (2014) adds to the SIFT algorithm by proposing a difference-of-Gaussians approach to increase efficiency to the point where the implementation can run in real-time. Fassold et al. (2015) further adds to SIFT by implements a GPGPU version using NVIDIA CUDA to further squeeze efficiency out of the algorithm. Given that SIFT and GLOH both perform well on accuracy metrics and that SIFT has been extended in multiple ways to be made more efficient SIFT is definitely the better algorithm for our project.

4.2 Background Subtraction

Background subtraction is a technique that detect movement between frames using a static camera [Piccardi 2004]. Background subtraction detects changes between the current frame and a reference frame. This is an effective means of detecting smooth motion given that the frames rate is above some threshold. Yilmaz et al. (2006) shows that the best fixed camera background subtraction algorithms, at the time of writing, were best applied to detecting regions of interest. In practice these algorithms did not perform well as object detection solutions but rather performed as feature analysis solutions finding regions of interest. Contradicting this the algorithms explored in Piccardi (2004) had varying degrees of speed, memory usage and accuracy where some algorithms were highly accurate when detecting moving objects. In Piccardi (2004), *Mixture of Gaussians* and *Kernel density estimation* both had very high accuracy. Adding to this *Running Gaussian average* was O(1) for each pixel for both memory and speed also scoring low-medium on accuracy making it a possible implementation for our project given that it can run in real-time. The paper presents an even spread of solutions based on speed, memory and accuracy requirements. Zheng et al. (2013) adds to the kernel density estimation technique that makes it highly parallelized and applicable to massive data set. This is a perfect option for our project given that it is very computationally efficient but can also handle the large pixel count of a 4k video stream.

4.3 Segmentation

Segmentation algorithms partition areas of a frame into similar regions [Yilmaz et al. 2006]. These partitions are based on a criteria or threshold value. Comaniciu et al. (2001) proposes a versatile and efficient mean-shift clustering algorithm with the possibility of fine-tuning the algorithm to fit many applications. This algorithm can be reliably scaled to perform edge detection as well. A graph partitioning system can also be used to segment frames [Yilmaz et al. 2006]. Using graph-cuts, a graph that represents partitions is pruned to create a number of disjointed areas that represent the segmentations. Shi et al. (2000) solves the common problem of bias towards over-segmenting a graph using a normalized cut implementation. This limits the number of cuts by creating a ratio that compares the weighted value of the nodes in the potential cut to the rest of the graph and prior cut segmentations. Segmentation algorithms create a contour representation of objects which is especially useful for our project as they can aid in interpreting hand gestures.

5. OBJECT TRACKING

Object tracking tracks detected objects across frames of a video [Yelmaz et al. 2006]. The tracking will culminate in a total trajectory of the object through the scene often using velocity and direction information to make intelligent conclusions where occlusion occurs. Common approaches include point tracking, kernel tracking and silhouette tracking.

5.1 Point Tracking

Point tracking uses the point representation for its objects [Yelmaz et al. 2006]. Point tracking is fazed mostly by the point correspondence problem caused by entry and exit of objects to the scene, false detections and partial or full occlusion. Veenman et al. (2001) proposes an approach that overcomes this problem using a heuristic that constrains the input data to a given number of interest points. Other constraints include proximity which expects objects to move incrementally between frames given that the time-step between frames is small [Yelmaz et al. 2001]. Maximum velocity limits the amount of movement an object can make between frames to a circle around the object. Small velocity change expects objects in the real world to incrementally change their velocity and direction. Common motion assumes interest points that are part of the same object (in the same region) will act similar to one another. Rigidity builds on common motion to expect objects to be rigid, thus the distance between interest points will remain the same. Sethi et al. (1987) propose an algorithm that uses proximity and rigidity to constrain the correspondence problem. Rangarajan et al. (1991) propose an algorithm that uses proximity and velocity change to constrain the correspondence problem. From these two papers it's clear that combinations of constraints mitigate the correspondence problem largely. Although it must be noted that Rangarajam et al. (1991) present an algorithm that doesn't handle entry and exit of objects furthermore occlusion and misdetection is only assumed rather than managed, making it less applicable to our project. Sbalzarini et al. (2005) implements an efficient point tracking algorithm with high accuracy that handles partial and full temporary occlusion of points. While this algorithm is applied to cell biology it performs video frame calculations on a 2D plane and is therefore applicable to this project especially given the occlusion handling of points.

5.2 Kernel Tracking

Kernel tracking on a single object computes the velocity and direction of movement of the object and uses a template shape to track the object [Yelmaz et al. 2006]. Templates used are primitives such as rectangles, circles and ellipses. Kernel tracking can be quite expensive especially if image intensity and color are used as features. Schweitzer et al. (2002) implements speedups of the algorithm by only searching locally around the kernel still rendering it too inefficient for our project. Gavrilla et al. (1998) proposes a template matching scheme that uses a hierarchy of templates to best match the object. A distance transform (DT) maps the distance of each pixel from the nearest feature pixel. The paper uses this by creating a binary template of the object of interest and comparing this to the calculated DT matrix. Tracking multiple objects can be implemented using the algorithm proposed in Tao et al. (2002). Their paper proposes splitting a frame into separate layers for the background and each of the objects being tracked. Each of the object layers has a template ellipse and calculates velocity and rotational information. Attributing pixels to a layer is calculated by comparing velocity and rotational information of the pixels. This provides lots of information to handle temporary occlusion. Zheng et al. (2013) propose a kernel density estimation that is efficient and can run on large data sets that is applicable to our project.

5.3 Silhouette Tracking

Silhouette tracking is best applied to objects that can't be accurately modelled using primitives geometric shapes [Yilmaz et al. 2006]. The human figure, for example, is most accurately tracked using silhouette or contour representations which yield lots of feature edge information. Silhouettes are most efficiently calculated using background subtraction but certain edge feature algorithms can also be applied efficiently to yield a contour representation similar to a silhouette. Shape matching is a technique implemented in Mori et al. (2005) that tracks silhouettes in a frame using a templated edge map generated from previous frames. The similarity of the template and the detected silhouette is then calculated to decide if they are the same object. Their algorithm is efficient because it prunes a large portion of the search space by retrieving a small subset of likely matching shapes before performing detailed matching calculations. Mondal et al. (2014) proposes an algorithm using a combination of local neighborhood information and fuzzy k-nearest-neighbor classification. The algorithm creates silhouettes based on a threshold of motion between frames and then applies contour tracking to silhouettes that pass this criteria. Their implementation runs quickly and has a higher stated accuracy than the implementation used in Mori et al. (2005). Having a contour representation for this project makes hand gesture recognition not only more viable but also more accurate.

6. CONCLUSSIONS

In this review paper we discussed multiple tracking algorithms as well as the division of object tracking into object representation, feature analysis, object detection and object tracking and how these portions relate. The overall goal of the paper is to decide on viable algorithms for our 4k lecturer tracking project. Important criteria for this is that the algorithm runs quickly and can track and lecturer and their gestures while also handling occlusion. For these criteria kernel tracking and silhouette tracking are decisively the best approaches. The algorithms from Zheng et al. (2013) and Mori et al. (2005) show the best promise. Kernel tracking implemented with a primitive representation would suffice but given that we need to track gestures as well as people the algorithm from Mori et al. (2005) using silhouettes transformed to contours fits best into our project requirements. Given that this algorithm uses contour representations hand gestures can also be tracked. Occlusion is a recurring issue that can mostly be handled with the use of velocity and orientation data by following a trajectory until the object is fully tracked again though some algorithms deal with occlusion better natively than others.

7. REFERENCES

[1] Ali, Anjum, and J. K. Aggarwal. "Segmentation and recognition of continuous human activity." *Detection and recognition of events in video, 2001. Proceedings. IEEE Workshop on*. IEEE, 2001.

[2] Ajoy Mondal, Susmita Ghosh and Ashish Ghosh, "Efficient silhouette-based contour tracking using local information", Soft Computing 2014. Pages 785 – 805. http://dx.doi.org/10.1007/s00500-014-1543-y

[3] Beauchemin, Steven S., and John L. Barron. "The computation of optical flow." *ACM computing surveys (CSUR)* 27.3 (1995): 433-466.

[4] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.5 (2002): 603-619.

[5] Comaniciu D., V. Ramesh and P. Meer, "Kernel-based object tracking," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-577, May 2003. doi: 10.1109/TPAMI.2003.1195991

[6] García J., A. Gardel, I. Bravo, J. L. Lázaro, M. Martínez and D. Rodríguez, "Directional People Counter Based on Head Tracking," in *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 3991-4000, Sept. 2013. doi: 10.1109/TIE.2012.2206330

[7] Gavrila, Dariu M. "Multi-feature hierarchical template matching using distance transforms." *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*. Vol. 1. IEEE, 1998.

[8] Han, Jungong, et al. "Enhanced computer vision with microsoft kinect sensor: A review." *Cybernetics, IEEE Transactions on* 43.5 (2013): 1318-1334.

[9] Hannes Fassold and Jakub Rosner. A real-time GPU implementation of the SIFT algorithm for large-scale video analysis tasks. Proc. SPIE 9400, Real-Time Image and Video Processing 2015, 940007 (February 27, 2015); doi:10.1117/12.2083201.

[10] Horn, Berthold K., and Brian G. Schunck. "Determining optical flow." *1981 Technical symposium east*. International Society for Optics and Photonics, 1981.

[11] Jain, A. and Zongker, D., "Feature selection: evaluation, application, and small sample performance," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153-158, Feb 1997. doi: 10.1109/34.574797

[12] Kim, Yeong-Taeg. "Contrast enhancement using brightness preserving bi-histogram equalization." *Consumer Electronics, IEEE Transactions on* 43.1 (1997): 1-8.

[13] Lerdsudwichai, Charay, Mohamed Abdel-Mottaleb, and A-Nasser Ansari. "Tracking multiple people with recovery from

partial and total occlusion."*Pattern Recognition* 38.7 (2005): 1059-1070.

[14] Lindeberg, Tony and K. Ikeuchi (ed.) (2014). Scale Selection. "Computer Vision: A Reference Guide" Pages 701-713 http://dx.doi.org/10.1007/978-0-387-31439-6_242

[15] Maini, Raman, and Himanshu Aggarwal. "Study and comparison of various image edge detection techniques." *International journal of image processing (IJIP)* 3.1 (2009): 1-11.

[16] Mikolajczyk, Krystian, and Cordelia Schmid. "A performance evaluation of local descriptors." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.10 (2005): 1615-1630.

[17] Milad Memarzadeh, Mani Golparvar-Fard, Juan Carlos Niebles, Automated 2D detection of construction equipment and workers from site video streams using histograms of oriented gradients and colors, Automation in Construction, Volume 32, July 2013, Pages 24-37, ISSN 0926-5805, http://dx.doi.org/10.1016/j.autcon.2012.12.002.

[18] Mitra, Sushmita, and Tinku Acharya. "Gesture recognition: A survey."*Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 37.3 (2007): 311-324. Fend Wang, Chong-Wha Ngo and Ting-Chuen Pong (2004). *Gesture Tracking and Recognition for Lecture Video Editing*. 17th International Conference on Pattern Recognition (ICPR'04).

[19] Mori Greg, Serge Belongie, Jitendra Malik, "Efficient Shape Matching Using Shape Contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 11, pp. 1832-1837, November, 2005

[20] Nguyen, Hieu T., and Arnold WM Smeulders. "Fast occluded object tracking by a robust appearance filter." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26.8 (2004): 1099-1104.

[21] Parks D. and L. Itti, "Integrating human context and occlusion reasoning to improve handheld object tracking," *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, 2014, pp. 436-440. doi: 10.1109/ICIP.2014.7025087

[22] Paschos, George. "Perceptually uniform color spaces for color texture analysis: an empirical evaluation." *Image Processing, IEEE Transactions on*10.6 (2001): 932-937.

[23] Piccardi, Massimo. "Background subtraction techniques: a review."*Systems, man and cybernetics, 2004 IEEE international conference on*. Vol. 4. IEEE, 2004.

[24] Phung, Son Lam, Abdesselam Bouzerdoum, and Douglas Chai. "Skin segmentation using color pixel classification: analysis and comparison."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.1 (2005): 148-154.

[25] Rangarajan, Krishnan, and Mubarak Shah. "Establishing motion correspondence." *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. IEEE, 1991.

[26] Sbalzarini I.F., P. Koumoutsakos, Feature point tracking and trajectory analysis for video imaging in cell biology, Journal of Structural Biology, Volume 151, Issue 2, August 2005, Pages 182-195, ISSN 1047-8477, http://dx.doi.org/10.1016/j.jsb.2005.06.002. (http://www.sciencedirect.com/science/article/pii/S1047847705001267)

[27] Schweitzer, Haim, J. W. Bell, and Feng Wu. "Very fast template matching."*Computer Vision—ECCV 2002*. Springer

[28] Serby, David, E. K. Meier, and Luc Van Gool. "Probabilistic object tracking using multiple features." *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 2. IEEE, 2004.

[29] Sethi, Ishwar K., and Ramesh Jain. "Finding trajectories of feature points in a monocular image sequence." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 1 (1987): 56-73.

[30] Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.8 (2000): 888-905.

[31] Smith, Joh R., and Shih-Fu Chang. "Automated binary texture feature sets for image retrieval." *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*. Vol. 4. IEEE, 1996.

[32] Sonka, Milan, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014. Alper Yilmaz, Omar Javed and Mubarak Shah (2006). *Object tracking: A survey*. Acm computing surveys.

[33] Szeliski, Richard. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[34] Tao, Hai, Harpreet S. Sawhney, and Rakesh Kumar. "Object tracking with bayesian estimation of dynamic layer representations." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.1 (2002): 75-89.

[35] Veenman, Cor J., Marcel JT Reinders, and Eric Backer. "Resolving motion correspondence for densely moving points." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.1 (2001): 54-72.

[36] Yan Zheng, Jeffrey Jestes, Jeff M. Phillips, and Feifei Li. 2013. Quality and efficiency for kernel density estimates in large data. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data* (SIGMOD '13). ACM, New York, NY, USA, 433-444. DOI=http://dx.doi.org/10.1145/2463676.2465319

[37] Yilmaz A., Xin Li and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1531-1536, Nov. 2004. doi: 10.1109/TPAMI.2004.96

[38] Yu Zhong, A. K. Jain and M. P. Dubuisson-Jolly, "Object tracking using deformable templates," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 544-549, May 2000. doi: 10.1109/34.857008