Proceedings of the 5th International Conference of Control, Dynamic Systems, and Robotics (CDSR'18)
Niagara Falls, Canada – June 7 – 9, 2018
Paper No. 130
DOI: 10.11159/cdsr18.130

A Survey of Hardware Advances and Techniques for Vision-Based Object Detection, Classification, and Tracking

Robert Selje II, Liang Sun
New Mexico State University
1040 South Horseshoe Street, Las Cruces, Unites States
rseljeii@nmsu.edu; lsun@nmsu.edu

Abstract **-** Mobile object tracking is a significant and challenging task in computer vision that plays a vital role in various applications, such as artificial intelligence, autonomous vehicles, medical imaging, robotics, and surveillance systems. The major contribution of this review paper includes the discussion of various hardware platforms used to acquire raw imagery information and implement computer vision algorithms in the phases of object detection, object recognition and classification, and object tracking.

Keywords**:** Computer Vision, Visual Sensors, Object Detection, Object Tracking, Object Classification.

1. Introduction

Computer vision is a field of study that duplicates human sight for computer and robotic systems to interact with an environment. The computer vision community has exploded in recent years and has found roots in various applications throughout different fields, such as collision avoidance [12, 46], robotics [11], security and surveillance [26], sensor fusion [41], target tracking [44, 45], traffic control [23], and unmanned aerial vehicles [19]. A major area within computer vision is the detection and tracking of moving objects. Although object detection and tracking has been widely studied, researchers are continuously developing more powerful algorithms as computer vision makes more of a presence in our daily lives.

While existing literature has summarized major techniques for detection, classification, and tracking of object, the unique contribution of this paper lies in the summary and analysis of various hardware platforms used to implement these techniques for objective detection, classification, and tracking.

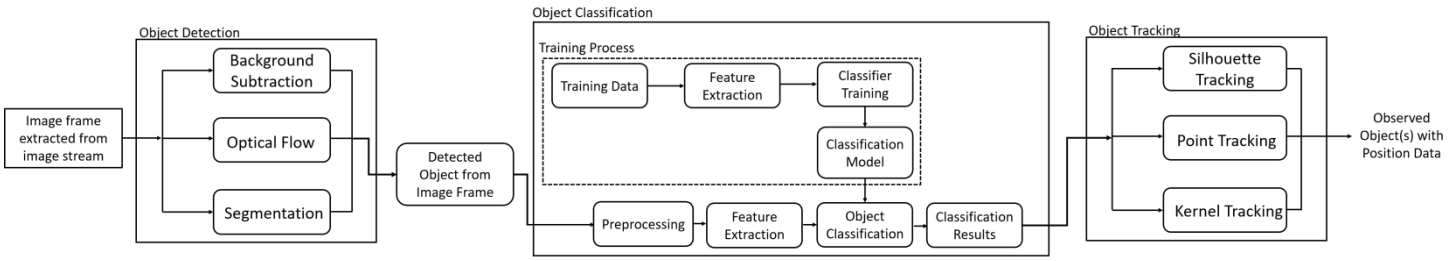

Fig. 1: Flowchart for detection and tracking of moving objects.

The detection and tracking of moving objects in image frames can be described as a recursive process of three steps, as shown in Figure 1. The first step, i.e., object detection, is to distinguish moving objects by identifying the pixels that have changed over consecutive image frames. Input for this step is a single image from extracted from a video stream captured from a hardware system. The second step, i.e., object classification, is to categorize each detected object in the frame as a class of features (e.g., a car, a human, or a robot). The third step, i.e., object tracking, is to follow the detected moving objects and create trajectories associated with corresponding objects.

The remainder of the paper is organized as follows. Section 2 discusses various hardware platforms used to capture images and to execute image processing algorithms. Section 3 presents the methods to detect moving objects in a video

frame. In Section 4, we discuss the detailed process of classifying detected objects. Section 5 compares different object-tracking algorithms and Section 6 concludes the paper.

2. Hardware for Computer Vision

Two types of hardware are discussed in this section. We first present the Graphics Processing Unit (GPU), a device that allows for onboard implementation of both image processing algorithms and advanced deep learning algorithms. Different sensing systems on the market are also presented that allows for capturing of objects in different environments.

2.1. Graphics Processing Units

Graphics Processing Units (GPUs), as shown in Figure 2(a), are a major innovation in the image processing field that allows for training and testing of algorithms in far less time than using Central Processing Units (CPUs) alone. GPUs also enable the training on much larger datasets to increase the accuracy of classifying objects in images, in far less time. GPUs contain thousands of small cores with a parallel architecture designed to handle raw throughput to compute millions of mathematical equations at one time. CPUs, on the other hand, contain a handful of cores designed for serial processing and are versatile to balance the load of tasks at one time. GPUs outperform CPUs in many technical aspects [22].

Most image processing algorithms are computationally demanding. In the past, onboard implementation of these algorithms can only be conducted using limited datasets with a small number of frames. Therefore, datasets were kept small to allow only one object type to be classified at a time. This kept the computation burden at a minimum, but limited the amount of research that could be done. Another challenge of onboard image processing was that these algorithms could not run in real time due to the required amount of computation it took to process each frame. Researchers used to select frames with distinct movements to help test tracking. With the use of GPUs, image processing algorithms now can be run in real time. In addition, more complicated video-processing algorithms [28] for object tracking needs substantial amounts of data to train the algorithm, which needs to be performed in a timely manner. GPUs limit the burden on the system by swiftly performing numerous redundant calculations in a parallel manner. This allows for quicker analysis of data and object classification.

(a) GPU Chip [4] (b) NVIDIA Drive[3] (c) LiDAR [5] (d) Infrared Camera [2] (e) Structured Light [1] (f) Stereo Camera [6]

Fig. 2: GPU Examples and Camera Types.

Efficiently combining the capabilities of GPUs and CPUs gives great potential for deep learning and analytic applications [40, 20]. Heavy processing workload of the CPUs can be offloaded by running heavy computational image processing calculations on GPUs and running general-purpose computations on CPUs. Companies, such as NVIDIA, are also producing dedicated hardware specifically for image processing. The NVIDIA Drive [3], as shown in Figure 2(b), is a powerful hardware device that contains two dedicated GPUs, allowing for approximately twenty-four trillion operations per second. This offers the capability to operate up to twelve cameras in real time.

2.2. Sensing System

Various types of sensing systems have been used to capture the visual world in the format of images at any time instantly. Four systems of interest are discussed in this section, including Light Detection and Ranging (LiDAR) systems, Infrared (IR) cameras, structured light systems, and stereo cameras, as shown in Figure 2(c-f) respectfully. A key characteristic of these systems is that they produce a video stream of continuous “image” frames that a computer system can extract at any given time instead of taking pictures at specific time intervals. Table 1 summarizes the pros and cons of these sensing systems.

2.2.1 LiDAR System

The LiDAR system [5] provides a highly accurate three-dimensional (3D) model of the surrounding environment including the height and distance of objects. The system operates by continuously sending out millions of laser beams per second and measures the time it takes for the reflected laser signal to arrive back at the sensor. Since the LiDAR system uses lasers to depict its environment instead of light for standard cameras, the resulting image lacks a color representation. The LiDAR can produce high resolution pictures regardless of the amount of ambient light present. This allows the LiDAR to operate at all times of the day, as well as areas containing little or no light. However, the LiDAR systems are subject to certain weather conditions, such as heavy rain, snow fog, and wind.

Table 1: Sensing Systems.

Camera Type	Operating Format	Advantages	Disadvantages
Light Detection and Ranging (LiDAR)	Laser	• Highly accurate 3D model • Height and distance measurements of objects • Ambient light is not needed	• Difficulty in heavy rain, snow and fog
Infrared camera (IR)	Heat	• Sense infrared radiation over large area for far distances • Sense through smoke, fog, dust, sand, and thin materials	• Difficulty seeing objects that vary erratically in temperatures • Cannot differentiate between objects in reflections on other surfaces.
Structure Light	Camera with light emitting source	• 3D pictures • Accurate 3D imaging • Low noise	• Short-range • Inefficient for some operations • Heavy data processing
Stereo Camera	Multiple cameras with different angles and focuses	• Wide field of view • Moving and still objects	• Less accurate 3D data • Heavy data processing

2.2.2. Infrared Camera

An Infrared camera [2] detects infrared radiation over a large area for far distances. Infrared radiation is a type of electromagnetic radiations that is invisible to human sight, but can be felt as heat. Infrared cameras are able to produce a visual representation of infrared energy in objects within the electronic spectrum range of 700 nm – 1 mm [32]. Infrared cameras can sense objects in any lighting condition, behind thin materials, and in an environment containing smoke, fog, dust, and sand. However, it has difficulty capturing objects that vary erratically in temperatures, as well as differentiating between objects with similar temperatures, or reflections on surfaces. Infrared cameras are sold with a specific range on the electromagnetic spectrum in mind. Two popular infrared cameras are the Near Infrared (700 nm to 1100 nm) [13] and Long Wavelength (8 μm to 14 μm) [16].

2.2.3. Structured Light System

The structured light system [1] is a 3D scanning device that is able to reconstruct a model by projecting different light patterns onto the scene and capturing the reflected light. The associated software uses the pattern from the camera to determine the 3D geometry through triangulation. The system is aware of the distance between the light emitting device and the camera, as well as, the angle at which the light emitted is pointing. With this information known, the system calculates the distance between the camera and the object. This allows the system to calculate the third dimension for a set number of pixels in the scene. This system works well for either moving or stationary objects. However, it does not provide very accurate 3D data and it is computationally demanding.

2.2.4. Stereo Camera

A stereo camera [6] is composed of multiple cameras at different angles and focuses to replicate binocular vision in human sight. This allows the camera to create images using all three dimensions [39]. It begins with identifying image pixels that correspond to the same point in a physical scene observed by multiple cameras. The 3D position of a point is then established by triangulation using a ray from each camera. The more corresponding pixels are identified, the more 3D points are determined with a single set of images. Correlation stereo methods [30] have been used to obtain correspondences for every pixel in a stereo image, resulting in tens of thousands of 3D vectors generated with every stereo image.

2.3. Integration of Multiple Sensing Systems

Each aforementioned sensing system uniquely represents the surrounding environment in either two or three dimensions and in either the RGB or greyscale color space. Research has been conducted [38] to combine multiple sensing systems with different field of view (FOV) representations in order to enhance the overall visual output.

Two distinct integrations have been reported, i.e., camera overlay [31] and camera networks [18]. Camera overlay is to superimpose the frames of different types of sensing systems in order to enhance the visual output; while a camera network expands the FOV of a sensing system by fusing the images captured by networked cameras at different perspectives. Singh et al. [38] presented the integration of the fields of view of a standard camera, also known as a visible light camera, and an infrared camera to produce a fused color image. The two overlaid images do not have to be equal in pixel size. Various techniques have been designed to fuse multi-resolution images together [29, 27].

A camera network consists of connected single cameras that expand the sensing coverage of the environment. Networked cameras are usually connected in two ways [26, 35]. The first way is through a technique called overlapping, in which cameras are closely placed together such that the images produced by neighboring cameras overlap with a certain portion. This allows for the creation of a continuously integrated image by stitching all individual image frames. Moving objects in any individual frame are captured easily when they move into another camera's FOV. However, the cost is considerably significant to construct such a large network of overlapping cameras. This is where disjointed camera networks come into play, in which cameras are strategically placed to maximize the observation of moving objects while limiting the possibility of losing the object in between neighboring cameras.

3. Object Detection

The object detection function, as shown in Figure1, works closely with the hardware to extract frames for analysis. Once the cameras are running, they produce a consistent stream of image frames containing the visual presentation of the environment within its FOV. These frames can be extracted at specific time intervals to track the objects found in the frame. The first step for object tracking is to identify the moving objects in each frame of a video sequence by grouping pixels of each object. We will analyze the three most common object-detection techniques, i.e., background subtraction, optical flow, and segmentation.

Background subtraction, also known as foreground detection, is a motion segmentation method used to detect moving objects in static scenes [9]. The idea behind background subtractions is to separate the background, consisting of all stationary items, from the foreground, containing all moving objects. Every image frame extracted from the video stream is compared to a reference frame containing only the background image. The two frames are compared by subtracting each pixel in the extracted image frame from the related pixel in the reference frame. If there are any differences between frames, the pixel from the extracted frame is considered moving and categorized as a foreground pixel.

Optical flow [7] defines the direction and the time rate for each pixel in sequential frames. This algorithm clusters pixels with similar characteristics together in order to approximate the displacement and velocities of the pixels between sequential frames. Each pixel in the frame is assigned a 2D velocity vector that is calculated based on the illumination changes over a set of sequential frames. The optical flow technique creates a vector field over time to detect the moving regions within an image as described by the object's movements from previous frames. Optical Flow is an accurate and effective algorithm, even with a moving camera. However, it is highly computationally demanding since calculations have to be conducted on each pixel with respect to both spatial and temporal coordinates. This requires special hardware to assist with the intensive computations to make it suitable for real-time video processing. It is also sensitive to noise due to its heavy dependency on clusters, and does not contain any method that deciphers a noisy pixel in a frame.

The image segmentation method partitions a frame into multiple sections by grouping together pixels of similar color, intensity, and texture [42]. Region detection method for segmentation divides a frame into a cluster of neighboring pixels that share the same information of color or intensity. The region detection methods extract all the pixels associated with an object from an image frame. The color information of each pixel is compared with each of its neighboring pixels. If the color difference between the two pixels are within a predefined range, then they are considered homogeneous and the region grows. Otherwise, the two neighboring pixels are considered as a boundary. The algorithm continues to check the pixels on the rim of the region until all pixels on the rim are confirmed as a boundary. The segmentation method represents all objects in the frame as its own entity, and it will then classifies and tracks each segment it finds. The segmentation method does not detect the presence of moving objects. However, the system can be trained to eventually recognize background pixels, but this process needs numerous frames and takes additional calculations to determine background pixels.

4. Object Classification

The object detection function produces an image frame that contains only the pixels of moving objects and is then passed onto the object classification function. Figure 1 shows a typical process for object classification using the Support Vector Machine approach. Before a system begins to classify objects from a live image stream, it must first be trained to look for particular objects. The training process uses a data set to extract basic features for the classifier training. The selection of data sets depends on the goal of applications and it largely affects the accuracy of object classification [15]. Typically, the training process first learns the basic features of 90% of the images in the category, and then uses the remaining 10% to validate the classifier. At the end of the training process, a classification model will be generated that classifies objects from a live video stream.

Each frame that is received from the object detection function has to go through a preparation phase (e.g., noise reduction, shadow removal, and object reorientation) before features can be extracted. Image noise refers to random disparity in a pixel's brightness or color. The Mean filter and Median filter are commonly used techniques to diminish noise in an image by reducing the intensity variation between pixels. Objects may also be scaled, rotated, or reflected about a point to have a proper orientation for subsequent processes [14]. Key features are then able to be extracted from the frame upon completion.

The feature extraction function generates a category of features in the format of interest points, i.e., a group of pixels that are easily detected in well-defined positions. Three state-of-the-art feature detectors include FAST, HOG, and SURF. The FAST [33] is a computational efficient algorithm used to detect corners in an image. Objects can be described in terms of the region around its corners. FAST is designed based on a circular mask segmentation test that allows the algorithm to perform with a high speed and accuracy, which enables feature extraction for real time video processing.

The appearance of an object is described by the distribution of its intensity gradients; the basic idea behind the Histogram of Oriented Gradients (HOG) algorithm. An image frame is divided into cells that contains the gradient magnitude of the pixels contained in the cell [43]. These gradient magnitudes are then used as a descriptor for the image. The gradient structure is characteristic of a shape within the frame. Therefore, the value of the concatenate will help uniquely identity the object.

The SURF method [8] uses a multi-resolution pyramid technique to blur the original image to detector blob features in an image. Each level of the pyramid is transformed using the Gaussian average to create a blurring effect and then scaled down. The idea behind the pyramid is that each pixel is the average for a set of pixels below it.

The final phase of the object classification phase is to run the extracted features through a Support Vector Machine (SVM). The SVM [37] is a highly efficient approach to classifying objects by plotting and grouping feature vectors in either two or three dimensional space. All the feature points learned during the training phase are plotted and then grouped together using hyperplanes, which act like boundaries that separate a set of feature vectors for particular objects. The SVM uses optimization techniques to calculate the median points between grouped feature vectors. When the system classifies objects, the objects' feature vectors are plotted and uses the hyperplanes to determine the group of learned feature vectors that it belongs to. The SVM provides a classification with a high degree of confidence.

An alternative method to SVN is a Convolution Neural Network (CNN) [24]; which is a deep learning algorithm for object classification by comparing pre-trained features with an input image pixel by pixel. This process is performed over multiple layers. Each layer breaks down an image into smaller pieces and creates a feature map of each piece at various

positions. CNN produces a very accurate prediction based on a weighted sum of the layers, which limits the amount of work required for preprocessing.

5. Object Tracking Using Multiple Cameras

The output of the Object Classification function, as described in Section 4, is a feature descriptor, i.e., is a set of feature points for each classified object. These feature points are used in the Object Tracking function, as shown in Figure 1, which refers to the process of locating moving objects in each frame of a video stream, generating a trajectory for each object, and estimating the path of objects as they move from frame to frame. Three popular techniques for tracking objects within a single camera includes: Point Tracking [10], Kernel Based Tracking [17], and Silhouette Based Tracking [34]. In the following section, we summarize recently-developed techniques for tracking using multiple cameras.

Three camera arrangements have been reported for objects tracking with multiple cameras: disjointed, overlapping, and the Pan-Tilt-Zoom (PTZ) [26]. Table 2 shows the advantages and disadvantages for each camera arrangement. It is critical that each camera informs other cameras in the network of the objects are in its FOV. The performance of this communication task depends on the design of the network architecture. A camera in the network either broadcasts the existence of objects in its FOV or works as a relay for specific neighboring cameras affected by the object's trajectory [21]. A popular network architecture is a centralized topology [36] where a sever handles all communication and processing burdens and frees up the cameras to only take video frame inputs. The server stitches the frames taken from all cameras and manages object tracking between cameras.

Table 2: Tracking with Multiple Cameras.

Style	Advantages	Disadvantages
Overlapping	• Object location is known at all times • Minimal communication between the nodes	• Limits the area that can by covered
Disjointed	• Covers a larger area	• Object location not always known • Additional communication between the nodes
Pan-Tilt-Zoom (PTZ)	• Cover a larger area with less cameras • Limited computation needed	• Limited with the number of targets to track

Object tracking with overlapping cameras [36] allows for a full coverage of an area where moving objects exist. Each camera in the network can track objects within its own FOV and store the information for detected moving objects. The object information is not shared between neighboring cameras until a moving object is in the overlapping region of two neighboring cameras. If a moving object leaves the FOV of a camera, that camera will know which camera to send the information for objects. This approach requires less computation and communication than a centralized network topology. However, the overlapping of FOVs limits the area that can be covered by the networked cameras.

Tracking objects with disjointed and spread-out cameras allows for objects to be tracked over a large area [18]. Additional computation and communication between cameras is needed. Since the FOVs of cameras do not overlap, there are uncovered "blind" regions between the FOVs of neighboring cameras, which causes temporary loss of tracking moving objects. The Kalman Filter is an effective method to estimate the object's location when it is in a "blind" region. However, the uncertainty of estimation can be very large when multiple "blind" regions are close to each other.

A pairing of cameras [25] that has been used for object tracking consists of a wide-view camera and a Pan-Tilt-Zoom (PTZ) camera, paired in a master-slave configuration. The wide-view camera, the master, remains stationary and sends commands to the PTZ camera, the slave, to track object(s) in the FOV of the wide-view camera.

6. Conclusion

This review paper provides a study of the object detection, classification, and tracking process, while summarizing a list of prevalent advanced hardware platforms and algorithms used within those stages. Various hardware platforms provide the computer vision system with an accelerated boost of raw throughput power while delivering a visual presentation of the surrounding environment using one or multiple sensing systems. While individual sensing systems and

techniques provides advanced capability of acquiring and processing image streams, techniques are needed to integrate these heterogeneous sources of information to provide an upgraded understanding of the environment.

References

[1] Basler, *Basler Time-of-Flight Camera*. [Online]. Available: https://www.baslerweb.com/fp-1507879569/media/downloads/documents/brochure/BAS1708_ToF_Brochure_EN_SAP0022_No5_web.pdf

[2] FLIR Systems, Inc., *FLIR Ax5-Series*. [Online]. Available: https://www.flir.com/globalassets/imported-assets/document/flir-ax5-usre-manual.pdf

[3] NVIDIA Corporation, *NVIDIA DRIVE Platform*. [Online]. Available: https://developer.nvidia.com/drive

[4] NVIDIA Corporation, *Nvidia GeForce GTX 780M: The New Best Graphics Card For Your Laptop*. [Online]. Available: https://www.geforce.com/hardware/notebook-gpus/geforce-gtx-780m

[5] Nanalyze, *Who is Velodyne and What is LiDAR?*. [Online]. Available: https://www.nanalyze.com/2016/08/who-is-velodyne-what-is-lidar/

[6] Stereolabs, *ZED - Depth Sensing and Camera Tracking*. [Online]. Available: https://www.stereolabs.com/zed/

[7] J. L. Barron, D. J. Fleet, S. S. Beauchemin, "Performance of optical flow techniques," *International journal of computer vision*, pp. 43-77, 1994.

[8] H. Bay, T. Tuytelaars, L. V. Gool, "Surf: Speeded up robust features," *Computer vision—ECCV 2006*, pp. 404-417, 2006.

[9] T. Bouwmans, F. E. Baf, B. Vachon, "Background modeling using mixture of gaussians for foreground detection-a survey," *Recent Patents on Computer Science*, pp. 219-237, 2008.

[10] F. Bu, Y. Cai, Y. Yang, "Multiple Object Tracking Based on Faster-RCNN Detector and KCF Tracker," 2016.

[11] Y. Cheng, *Object Recognition on iCub Robot*. 2017.

[12] C. Cigla, R. Brockers, L. Matthies, "Image-based Visual Perception and Representation for Collision Avoidance," *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 421-429, 2017.

[13] C. Fredembach, *Near-Infrared Imaging*.

[14] P. M. Corcoran, C. Iancu, *Automatic face recognition system for hidden markov model techniques* in *New Approaches to Characterization and Recognition of Faces*. InTech, 2011.

[15] T. Corradi, P. Hall, P. Iravani, "Object recognition combining vision and touch," *Robotics and Biomimetics*, pp. 2, 2017.

[16] S. Gunapala, S. Bandara, J. Liu, C. Hill, S. Rafol, J. Mumolo, J. Trinh, M. Tidrow, P. LeVan, "1024 ×1024 pixel mid-wavelength and long-wavelength infrared QWIP focal plane arrays for imaging applications," *Semiconductor Science and Technology*, pp. 473, 2005.

[17] R. K. Jatoth, S. Shubhra, A. Ejaz, "Performance comparison of Kalman filter and mean shift algorithm for object tracking," *International Journal of Information Engineering and Electronic Business*, pp. 17, 2013.

[18] O. Javed, Z. Rasheed, K. Shafique, M. Shah, "Tracking across multiple cameras with disjoint views," *Proceedings of the Ninth IEEE International Conference on Computer Vision*, vol. 2, pp. 952, 2003.

[19] S. Kamate, N. Yilmazer, "Application of object detection and tracking techniques for unmanned aerial vehicles," *Procedia Computer Science*, pp. 436-441, 2015.

[20] N. Kasmi, M. Zbakh, S. A. Mahmoudi, P. Manneback, "Performance Evaluation and Analysis for Conjugate Gradient Solver on Heterogeneous (Multi-GPUs/Multi-CPUs) platforms," *IJCSNS*, pp. 206, 2017.

[21] S. Khan, O. Javed, Z. Rasheed, M. Shah: "Human tracking in multiple cameras," in *Proceedings of Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001*, pp. 331-336, 2001.

[22] M. Kim, H. Shin, J. Jung, S. Kim, D. Yoon, T. S. Suh, "Development of GPU-based fast reconstruction algorithm for Gamma ray imaging with insufficient conditions," 2017.

[23] D. Koller, J. Weber, J. Malik: "Robust multiple car tracking with occlusion reasoning," pp. 189-196, 1994.

[24] J. Li, X. Zhou, S. Chan, S. Chen, "Object tracking using a convolutional network and a structured output SVM," *Computational Visual Media*, pp. 1-11, 2017.

[25] Y. Lu, S. Payandeh, "Cooperative hybrid multi-camera tracking for people surveillance," *Canadian Journal of Electrical and Computer Engineering*, pp. 145-152, 2008.

[26] E. Monari, J. Maerker, K. Kroschel, “A robust and efficient approach for human tracking in multi-camera systems,” *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009. AVSS'09*, pp. 134, 139, 2009.
[27] G. Pajares, J. M. De La Cruz, “A wavelet-based image fusion tutorial,” *Pattern recognition*, pp. 1855-1872, 2004.
[28] H. Pawar, K. Darekar, P. Paliwal, P. Darak, S. Tiwaskar, “Survey on Object Detection from Video Sequence,” *International Journal For Research In Emerging Science And Technology*, 2014.
[29] V. S. Petrovic, C. S. Xydeas, “Gradient-based multiresolution image fusion,” *IEEE Transactions on Image processing*, pp. 228-237, 2004.
[30] L. Pérez, Í. Rodríguez, N. Rodríguez, R. Usamentiaga, D. F. García, “Robot guidance using machine vision techniques in industrial environments: A comparative review,” *Sensors*, pp. 335, 2016.
[31] K. Rani, R. Sharma, “Study of different image fusion algorithm,” *International journal of Emerging Technology and advanced Engineering*, pp. 288-291, 2013.
[32] A. Rogalski, “Infrared detectors: an overview,” *Infrared Physics & Technology*, pp. 187-210, 2002.
[33] E. Rosten, T. Drummond, “Machine learning for high-speed corner detection,” *Computer Vision—ECCV 2006*, pp. 430-443, 2006.
[34] R. K. Rout, *A survey on object detection and tracking algorithms*. 2013.
[35] A. S. Samdurkar, S. Kamble, N. Thakur, A. S. Patharkar, “Overview of Object Detection and Tracking based on Block Matching Techniques,” pp. 313-319, 2017.
[36] S. A. Sagar, A. Holambe, “A Noval System Architecture for Multi Object Tracking Using Multiple Overlapping and Non-Overlapping Cameras,” *International Journal of Biotechnology and Biochemistry*, pp. 275-283, 2017.
[37] I. Setitra, “Object classification in videos-an overview,” *Journal of Automation and Control Engineering*, pp. 106-109, 2013.
[38] S. Singh, A. Gyaourova, G. Bebis, I. Pavlidis, “Infrared and visible image fusion for face recognition,” *Proceedings of SPIE*, pp. 585-596, 2004.
[39] S. Teja, S. Asif, K. Bhavani, B. Charan, B. Phaneendra, “Stereo Vision,” *International Research Journal of Engineering and Technology (IRJET)*, pp. 636-639, 2017.
[40] G. Teodoro, R. Sachetto, O. Sertel, M. N. Gurcan, W. Meira, U. Catalyurek, R. Ferreira, “Coordinating the use of GPU and CPU for improving performance of compute intensive applications,” *Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on*, pp. 1-10, 2009.
[41] K. R. Yunus, M. Mechkul, “Multiple Sensor Fusion for Moving Object Detection and Tracking,” *International Research Journal of Engineering and Technology (IRJET)*, pp. 1214-1217, 2017.
[42] N. M. Zaitoun, M. J. Aqel, “Survey on image segmentation techniques,” *Procedia Computer Science*, pp. 797-806, 2015.
[43] Q. Zhu, M. Yeh, K. Cheng, S. Avidan, “Fast human detection using a cascade of histograms of oriented gradients,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006*, pp. 1491-1498, 2006.
[44] L. Sun, and D. Pack, "Guidance Law Design for Tracking Mobile Ground Targets Using An Unmanned Aerial Vehicle with A Fixed Camera," *IEEE International Conference on Unmanned Aircraft Systems*, Arlington, VA, USA, Jun. 2016, pp. 235-241.
[45] L. Sun, and D. Pack, "Mobile Target Tracking Using an Unmanned Aerial Vehicle with a Non-Gimbaled Video Sensor," *AIAA Science and Technology Forum: AIAA Guidance, Navigation, and Control Conference*, Kissimmee, Florida, USA, Jan. 2015.
[46] J. Lwowski, L. Sun, R. Mexquitic, R. Sharma, and D. Pack, "A Reactive Bearing Angle Only Obstacle Avoidance Technique for Unmanned Ground Vehicles," *Journal of Automation and Control Research*, vol. 1, no. 1, pp. 31-37, 2014.