Automated Tracking and Real Time Following of Moving Person for Robotics Applications

Submitted: 2nd April 2018; accepted: 20th December 2019

Jignesh J. Patoliya, Hiren K. Mewada

DOI: 10.14313/JAMRIS/4-2019/35

Abstract: *Presently the interaction of robots with human plays an important role in various social applications. Reliable tracking is an important aspect for the social robots where robots need to follow the moving person. This paper proposes the implementation of automated tracking and real time following algorithm for robotic automation. Occlusion and identity retention are the major challenges in the tracking process. Hence, a feature set based identity retention algorithm is used and integrated with robot operating system. The tracking algorithm is implemented using robot operating system in Linux and using OpenCV. The tracking algorithm achieved 85% accuracy and 72.30% precision. Further analysis of tracking algorithm corresponds to the integration of ROS and OpenCV is presented. The analysis of tracking algorithm concludes that ROS linking required 0.64% more time in comparison with simple OpenCV code based tracking algorithm.*

Keywords: *Visual tracking, Robot operating system, identity retention, Gaussian Mixture Model*

1. Introduction

Robots have become an integrated part of the human and cooperation between the human and robot seems to be a promising way to increase the productivity and flexibility. The new open source software like robot operating system (ROS) has established the convenient way of use of robotics accessible in both research and consumable applications. Object tracking is one of the important steps for the applications requiring recognition, monitoring and communications with human. And the challenging task is to recognize and track human from the group of people. The motion of the human and occlusion are the common causes for false detection and lost tracking. Secondly, to share the physical environments between the human and robots, there is a need to avoid collision with robots [1]. If movement of human is detected by the robot then this information can be used by the robots to prevent the collision. For robotics applications, various sensors have been used to detect the human like laser sensor to detect the movement of leg, RGB camera, RGB-D (visible band images with depth information), ultrasonic sensor, proximity sensor etc. Proposed application needs identification of the desired person from the group as well as tracking of that person. Therefore, looking to the complexity and with consideration of quantitative parameters like accuracy and range of detection camera based tracking is most suitable in comparison with other available sensor. This paper proposes the implementation of human tracking algorithm linked with ROS for robotic applications in visible band.

Vast research is carried out till date for tracking humans in computer vision applications. Major constraints involved in the implementation of tracking algorithm for robotics applications are the performance of sensors, illumination variation, shadow, occlusion and computational complexity to perform in real time. However, few considerations have been given to resolve occlusion problem in real time scenario for ROS applications. This paper proposes the real time tracking along with crucial occlusion handling for robots such that robot do not miss the actual target among the group of people to be followed. With this objective the section II discuss the previous work about the tracking which has been targeted for robotic applications specifically. The section III represents constrains involved in the ROS based tracking and it is partitioned into two parts.

First part discusses the adopted tracking algorithm for real time video application and second part discusses the integration of this tracking algorithm with ROS environment. In Later section, evaluation of the proposed tracking algorithm and conclusion are presented.

2. Literatures Surveys in ROS and Tracking

ROS provides the huge support for computer vision applications allowing integration of OpenCV libraries and tools. The availability of the high-end camera suitable to ROS makes it easy to characterize and detect the human. Generally five approaches have been proposed in various literatures. This includes detection of human from depth images (i.e. uses of Microsoft Kinect sensor), tracking in visible band images (i.e. RGB images), Face detection based tracking, head position and torso estimation based tracking and leg detection based tracking (i.e. Laser range detection model). In addition, tracking applications are classified either as indoor tracking or outdoor tracking (i.e. surveillance applications).

Blob detection using GMM to detect number of objects in the frame has been discussed in many literatures [2, 3] and even segmentation of objects with its shape is also presented by Hiren et al [4] using Zynq processor. MATLAB is one of the powerful tools for computer vision applications. In [5], various human detection & tracking algorithms developed and implemented using ROS and MATLAB16B. Decision tree based machine learning technique is implemented to predict human activity. Classification of human activity is performed using Machine learning algorithms like k-nearest neighbours, multi class SVM and Decision Trees which gave a prediction accuracy of 96.02% on the test set.

In [6], authors discussed the issues in fully automated outdoor surveillance systems. In a realistic scenario, object tracking in a single camera is performed using background subtraction followed by region correspondence. They proposed recurrent motion Image method to carry out object detection by solving the problem of shadows and by handling spurious objects in the classification process. This system was tested on real-time video at approximately 15Hz for the colour images of 320x240 resolutions and proved to be computationally efficient.

A project work to measure the orientation of the object and to track its position on 3-dimension space was carried out by J. Diaz et al [7]. In this work,the orientation of human hand is measured by tracking it at shorter range. A leap motion sensor has been used to track the human hand. They presented two approaches, firstly, uses of a feature based computer vision algorithms and secondly, uses of a contour based particle filter to track and determine the orientation. They obtained promising results in hand tracking and gesture recognition which can be used human-interaction applications. To track the articulated objects, a visual feedback technique using machine learning algorithm was proposed in [8]. They developed a model to control the robot motion using the obtained estimation as feedback signal. They conducted the experiment using HRP-4 humanoid robot.

A ROS based robot is developed to track the trajectory in [9]. A 2D map of the environment was required for localization of the robot. The author proposed tracking of two different trajectories using a robot that follows nonholonomic constraints.

In [10], a vision based tracking of a moving target is proposed. Two quad-rotors, one as a target and another as a tracker is used. Tracker is equipped with various sensors such as camera and LIDAR to track the target. The camera feed is converted into frames and is processed using an image processing algorithm based on target detection. The trajectory of the moving target is then estimated using a coordinate transformation algorithm. Based on the estimated trajectory, the tracker moves itself in order to capture the new frames of the tracker. The complete model is developed in ROS and is evaluated on the basis of real-world frames.

A framework for the human following robot was proposed by Priyandoko et al. [11]. They used RGB-Depth images and laser sensor in the implementation. They developed four tracking models using the face, leg, colour and blob for the indoor environment. The implementation is constraints for single human only. The concluded that face detection need front face to identify which is not possible for human following robot application, whereas leg detection is unreliable. Colour detection is efficient as it need low processing. However, the approach is not applicable in an outdoor environment. Hence, this paper proposes the algorithm based on blob detection which is computationally efficient and fast.

3. Proposed Model

3.1. Constraints for Real Time Tracking

The goal of the paper is to make robot smart to track the person in real time, the tracking methods offering low inference speed are considered. The algorithms which use the use appearance features i.e. face, histogram of gradient features, mean-shift algorithms are relatively slow in computation and hence they cannot be used for the real-time tracking [12]. Instead, statistical, geometrical and positional features offering fast computation can be used. Lastly, paper uses and combine ideas from existing methods and attempt to implement them in an efficient way.

Real time object tracking need automated detection of object of interest from the captured camera video. The object can be persons, animals, vehicles, etc. Inorganic objects like tress, also referred as scene structure [13] may create confusion in identification of desired objects. Though background subtraction has few limitations i.e. fails to segment object under color characteristic variation and fails to discriminate between the shadow and desired object, due to its fast processing it has been used to detect the moving object in the proposed algorithm.

Robot detects a multiple moving people from the group of the people. However, it has to follow the specific people from this group. Hence, it has to track the people to whom it has been targeted. This makes it very challenging as it has to discriminate the desired person from the group and if any occlusion occurs, then it has to maintain identity of real person to follow. Where, occlusion is referred as overlapping of two objects and hence object disappears in the frame as it is blocked by other object. Various occlusion types were discussed in [14] like partial occlusion where all key features of the objects to be tracked are not available, full occlusion where object to be tracked is completely behind other object and long term occlusion. All These occlusion restricts the accurate position measurement of the desired person. The robot fails to recognize the correct person amongst the group of the people due to this occlusion. This paper proposes the implementation of ROS integrated tracking application where feature descriptions are used to discriminate the real person from the group of the people and to maintain its identity under occlusion.

3.2. Implementation of Person Tracking and Occlusion Handling Using Features Sets

Person Detection:
The very first step in the tracking is the foreground object detection. The standard Gaussian mixture model (GMM) based foreground detection method is used. In GMM, the probability of each pixel is calculated using the equation 1 as follows [2]:

$$P(Xi)=\sum_{i=1}^{K} W_{i,t}\eta(X_t,\mu_{i,t},\sigma_{i,t}) \quad (1)$$

Where K represents the number of distributions, $W_{i,t}$ is weight associated with i^{th} Gaussian at time t with mean $\mu_{i,t}$ and variance $\sigma_{i,t}$. η is the probability function given as

$$\eta(X,\mu,\Sigma)=\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}e^{-\frac{1}{2}(X_t-\mu)\Sigma^{-1}(X_t-\mu)} \quad (2)$$

This GMM provides number of blobs corresponds to detected number of people in the given frame with subtracting the background. Each blob is represented by rectangle window covering the detected area of the person. The morphological operations are performed to separate the actual blobs from the noise blobs. The goal for the tracking the human is to assign the identity to each detected the person and track to them. To assign the identity (ID), centroid of each detected blob is calculated and corresponding IDs are assigned to each blob.

This paper follows the tracking algorithm as expressed by Israni and Mewada [3]. The Hungarian cost function is calculated to track the person. It offers an advantage of finding an optimal object from the given sets in polynomial time. Thus for real time object tracking it is one of the best suitable methods. The major constrains involved in the Hungarian Cost matrix calculation is true blob identification. Therefore, to avoid the false identification of the person from the set of blobs, the blob area is calculated and compared with a predefined threshold. If calculated area exceeds this threshold then only it will be counted as person blob and it will be tracked in the neighbor frame. Due to variation in the video format (i.e. size of the frame contained by the video), the threshold also varies with the type of datasets. Using the trial and error method, it has been found that the threshold is related with the frame size and it can be calculated as.

$$\tau=\frac{R\times C}{100} \quad (3)$$

Where τ represents the threshold, R and C are the number of rows and columns of the frame of video. Figure 1 [3] proposes the example of the true blob detection related to the person for the video dataset consists of 576 X 768 frame size. To identify the true blob (i.e. presence of human in captured frame), detected blob from GMM is calculated over the number of frames. As shown in figure 1, one of the frames contains the blob area of 663 pixels. If same blob area is observed over the consecutive next frames, it has been found that area is increasing and it forms the shape identical to human pose. Thus using the equation 3, as it exceeds the threshold value of 4423, it can be counted as true blob. Same algorithm is repeated for all blobs obtained from GMM for each frames.

Fig. 1. True blob detection from the frame

Occlusion detection:
The major challenge in tracking process is occlusion which causes the invisibility of the desired person to be tracked. There are two causes for the invisibility of the desired person. Firstly, there is no movement of the person to be tracked. This happened because in the tracking method, GMM is used and GMM has a limitation that it can detect only moving person. Using the bounding box over the detected blobs, Kuhn–Munkres algorithm has been used to match blob between two consecutive frames. This algorithm calculates matching score of bounding box within two consecutive frames as expressed in figure 2.

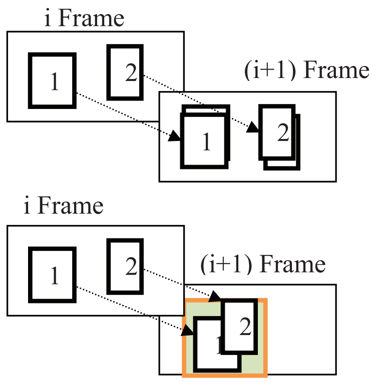


Fig. 2. Occlusion of two objects: matching score of bounding boxes in two frames (top), normal tracking without occlusion (bottom)

Figure 2 represents two cases. Firstly, there is no occlusion in next frame for two object. In this case, objects moves with linear displacement in consecutive frames. And the displacement of the object is independent of other objects and motion of the camera. The present state of each object can be modeled as

$$X_i=[Cx_i,Cy_i,A_i,s_i,\Delta Cx_i,\Delta Cy_i,\Delta A_i,\Delta s_i]$$

Where Cx_i and Cy_i represents the pixel location of blob's center, A_i and s_i give area and span of the blob window. Rest parameters are corresponding velocity displacement. Without occlusion, each state parameters can be predicted without any correction. This estimation model of Kalman filter can easily track objects in the video.

Secondly, first object is occluded by second object and hence GMM will detect single object having large blob area (i.e. shaded region) in comparison. Major constraints in tracking is occlusion. The best condition to identify the occlusion (i.e. occluder and occluded person) is that number of the detected blobs in the current frame is less than the number of blobs detected in the previous frame and the area of the blob having occlusion (i.e. are of bounding box) is always larger than the blob of a single person.Further occlusion can be classified as self-occlusion where part of person overlap itself, inter-object occlusion due to overlapping of two persons.

Self-occlusion can be detected easily by counting number of bounding boxes and number must be equal for two consecutive frames. The mismatch of number of bounding boxes between the frames represents the inter-object occlusion. This can be detected by comparing the aspect ratio of two bounding boxes. Let $AR_i = s_i^2/A_i$ and $AR_{i+1} = s_{i+1}^2/A_{i+1}$ are the aspect ratio of bonding boxes in two consecutive frames. If aspect ratio is greater than 0.5 then it depict the inter-object occlusion i.e.

$$Occl = \begin{cases} 1 & AR_i > 0.5 \text{ or } AR_{i+1} > 0.5 \\ 0 & else \end{cases} \tag{4}$$

This process is repeated for all detected blobs (i.e. persons) in the current frame.

Identity retention process:

The important aspect of ROS based tracking algorithms not only to detect the occlusion, but to maintain the identity of all detected person such that robot can tracked the desired person from the group of the people. Thus during the demerging process, the preservation of identity is must.

To preserve assigned identity to each person, features based matching algorithm is used. The aim is that robot should detect and track the person in real time. Hence, unsupervised model with mutual comparison the statistical features of the detected persons is used in the algorithm. The set of features including central moments, derivative of Gaussian, steerable filter based features, Gray level local variance, energy of the gradient, discrete cosine transform features, Laplacian filter based edge features of the person under occlusion are used as presented in [3]. Let F_{cp_1} represent features set of the desired person having identity of P_1 and F_{cp_2} is the features set of the person who occluded *P1* and assigned identity is P_2. Similarly, F_{rp_1} and F_{rp_2} are features sets of person P_1 and P_2 in the reference frame where both are separate and identifiable. The identity assignments during the demerging process are calculated as follows:

$$S_1 = \sum_i \left| F_{cp_1}(i) - F_{rp_1}(i) \right| + \sum_i \left| F_{cp_2}(i) - F_{rp_2}(i) \right| \tag{5}$$

$$S_2 = \sum_i \left| F_{cp_1}(i) - F_{rp_2}(i) \right| + \sum_i \left| F_{cp_2}(i) - F_{rp_1}(i) \right| \tag{6}$$

Using the comparison, if S1< S2 then P_1 will be assigned to the blob having features set of F_{cp_1} and P_1 will be assigned to the blob having features set F_{cp_2} or assigned identity will be changed accordingly. Similar process is extended to detect the multiple objects and to maintain the identities of multiple occluded objects. For present robotic application, robot needs to track desired person only. So the algorithm will keep the identity of the desired person by discarding other detected identities. This identity of the tracked person along with position (i.e. (x, y) position and width and height of the blob) are published to the ROS node.

The algorithm is tested for various videos including PETS2009-S2L1 [15] and town Centre datasets [16]. Overall flowchart of the proposed model is presented in figure 3.

The tracking results among the few frames where occlusion occurs and algorithm handle the identity retention are presented in figure 4. It shows that person with assigned identity of 20 is being occluded by another person having identity of 2. And algorithm maintains the identity of both persons in demerging process of both the persons.

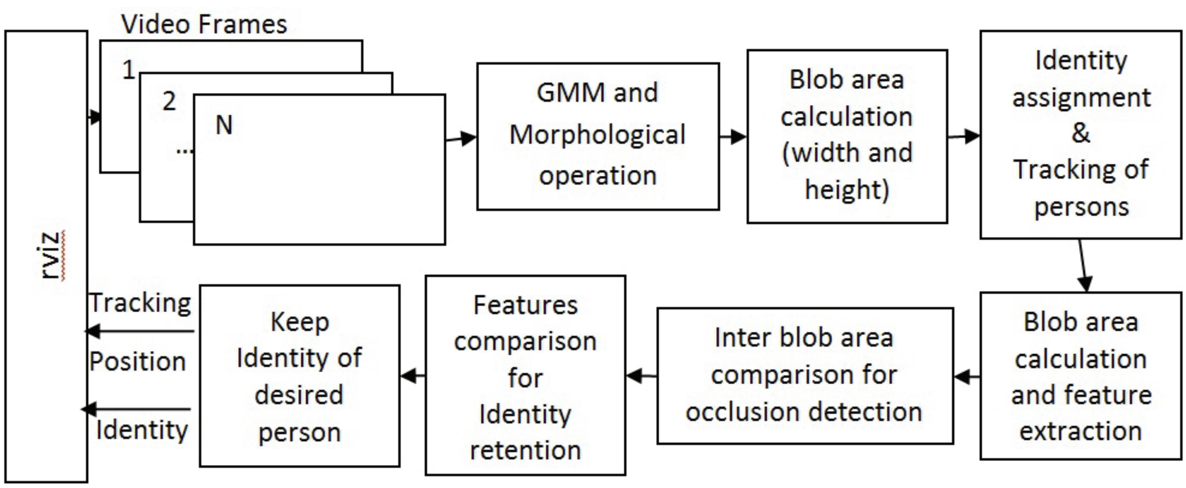
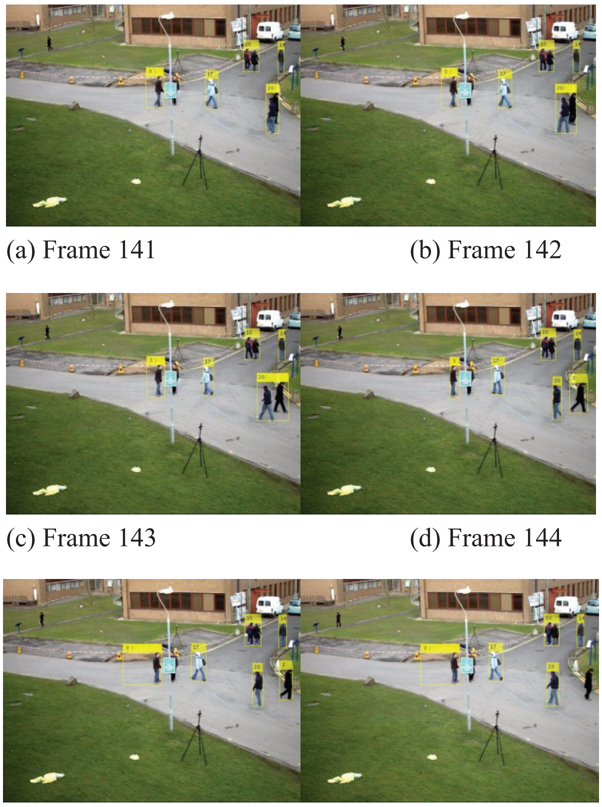


Fig. 3. Person identification and tracking algorithm

3.3. Linking of Tracking Algorithm with ROS

To fulfil the need of emerging robotics applications, the community has developed a common framework entitled Robot Operating System (ROS). It supports multiple programming languages. The proposed model is implemented using OpenCV. Entire implementation is carried out using ROS kinetic with Ubuntu 16. In ROS, all the software modules are represented as nodes. These nodes communicate using an environment provided by a process called ROS Core. A variety of communication mechanisms are available in ROS. The one which is used in the proposed model is Publish – Subscribe Mechanism. It is an asynchronous mechanism in which two nodes can communicate using a pre-defined topic.



(a) Frame 141 (b) Frame 142

(c) Frame 143 (d) Frame 144

(e) Frame 145 (f) Frame 146

Fig. 4. Occlusion handling and identity retention process in frames of PETS2009-S2L1 video

One of the nodes has to adopt the role of publisher which sends the message, and another node has to adopt the role of subscriber to receive the message. The subscriber node subscribes to a topic and then waits for the message to arrive. Whenever the publisher node sends a message on the predefined topic, the subscriber node will get a callback with the received message. In the proposed implementation, ROS package of GMM [17] is used in the implementation. It finds optimum number of Gaussian components using Bayesian approach. Then morphological operation is used to process the output obtained from this GMM. This removes the noise and true blob as explained in section III (b). A publisher node is created to connect with the camera and fetch frames from it. These frames are processed to create blob areas as explained earlier. The blob area along with its width and height are converted to custom ROS messages using cvBridge library. The tracking algorithm also generates the identity for all the detected people. These identities are also appended in the previously generated ROS message. Finally, this ROS message is passed on to the second node using the Publish-Subscribe mechanism. Custom messages are generated using ROS message utility. The subscriber node accepts the data sent by Publisher node and generates the tracked output. The subscriber node has two functions, one to instruct actuator to follow the person based on the message it received and second to display the image of the tracked person.

For the second functionality, the subscriber node converts ROS message back into vector which is compatible with OpenCV and then it is displayed. The ROS message formation and communication are presented in figure 5 where region inside the dotted rectangle presents the ROS environment and outer region represents the interfacing of peripheral components.

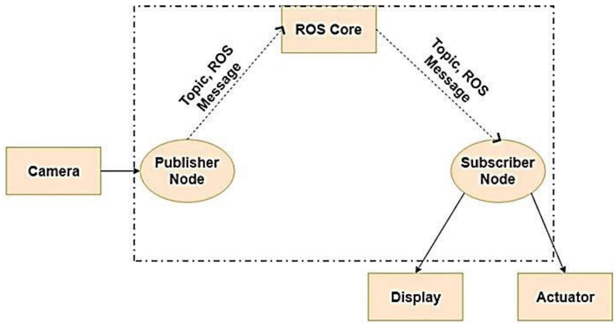


Fig. 5. Conceptual diagram of ROS message formation and communication

4. Evaluation and Discussion of the Tracking Algorithm

Initially, the algorithm is evaluated on two datasets of PETS2009-S2L1 and town center. The data-sets contain videos with brightness adaption and occlusion. Frame rate for the videos is 25 frames per second. The direction of the person changes abruptly and randomly. This is one of the challenging tasks and this scenario fits to the robotic applications. The algorithm's detection is evaluated by comparing the number persons detected by algorithm with actual number of persons to be detected (i.e. ground truth datasets). This can be calculated using intersection of union (IoU). Which measures overlap of two bounding box. Figure 6 represents accuracy calculation where True Positive (TP) represents actual overlap region of detected and ground truth object, False positive (FP) represents the detected object which is not part of ground truth object and False negative (FN) gives region which is not detected by the algorithm.

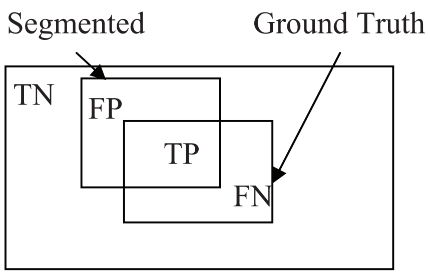


Fig. 6. Object detection Accuracy measurement using dice similarity coefficient

Using these parameters, accuracy is defined as

$$Acc(\%) = \frac{2TP * 100}{(TP + FP) + (TP + FN)} \tag{7}$$

The object detection accuracy graph for each frame in the video is presented in figure 7.

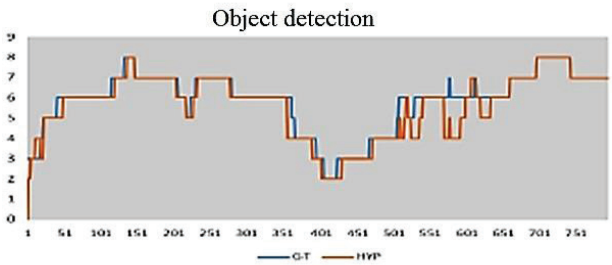


Fig.7. Accuracy of object detection in the video

The evaluation of tracking algorithm includes calculation of Multi object tracking accuracy (MOTA) and Precision (MOTP) [3].Where, MOTA calculates algorithms accuracy to detect and track the persons in the video, and MOTP calculates the positional accuracy of the tracked person in the video. They can be calculated using following equations 8 and 9.

$$MOTA = 1 - \frac{\sum_t m_t + FP_t + mme_t}{\sum_t g_t} \tag{8}$$

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_t c_t} \tag{9}$$

Where, m_t is number of misses, mme_t is number of mismatch, $d_{i,t}$ is distance between the object i and its corresponding object in ground truth set and c_t gives number of found matches in given time t. It achieved 85% and 70% MOTA and 70.70% and 72.30% MOTP.

Further analysis has been carried out to see the conversion time and speed of the OpenCV and its linkage with ROS. The figure 8 presents the execution time comparison for the program compile using OpenCV only and when program is linked with ROS for six videos. Average execution time for to process entire video and provide the output message of location of identity is 61.95 sec. It proposes that linking of ROS hardly invokes approximately 0.64% execution time.

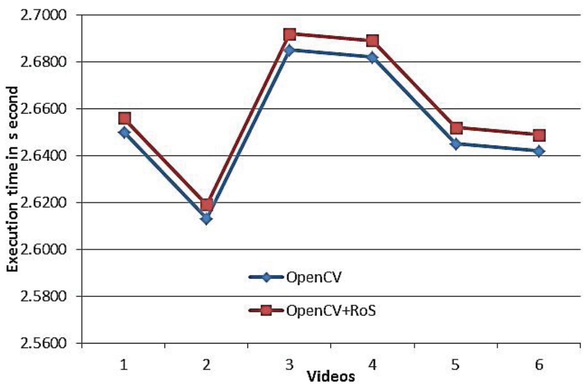


Fig. 8. Video execution time comparisons for OpenCV and inclusion of ROS with OpenCV

5. Conclusion

Social robots are taking the role to interact with human for various social service requirements like hospitals, airports, museums, etc. Where, social robot helps human by following him. To fulfil this requirement, a reliable tracking is required. This paper attempts to develop reliable tracking by conquering the challenge of occlusion and identification of the person to be followed from the group of people. For real time application, the fast detection and identification of the desired person is required. Hence, GMM based person tracking is used in the proposed model. For fast and accurate identification of the desired person, a set of features which are rotational and scale invariant for efficient detection and identity retention is used. The algorithm founds to be 85% accurate in detecting the number of persons with 72.30% precision rate. All implementation has been done using ROS and OpenCV. Analysis of linking of OpenCV with ROS is also expressed in the study and it has been observed that ROS addon 0.64% time in the OpenCV code. In feature, a prototype model will be developed. As part of software testing, architecture can be further revised such that it process parallel processing of tracking and identity retention process.

AUTHORS

Jignesh J Patoliya* – Electronics and Communication Engineering Department, Chandubhai S Patel Institute of Technology, Charotar University of Science and Technology, Post Changa, Gujarat, India, email: jigneshpatoliya@charusat.ac.in.

Hiren K Mewada – Assistant Research Professor, Electrical Engineering Department, Prince Mohammad Bin Fahd University, Kingdom of Saudi Arabia, email:mewadahiren@gmail.com.

*Corresponding author

REFERENCES

[1] M. Munaro, C. Lewis, D. Chambers, P. Hvass, E. Menegatti, "RGB-D Human Detection and Tracking for Industrial Environments". In: *Intelligent Autonomous Systems 13*, 2016, 1655–1668 DOI: 10.1007/978-3-319-08338-4_119.

[2] D. Reynolds, "Gaussian Mixture Models". In: *Encyclopedia of Biometrics*, 2009, 659–663.

[3] D. Israni, H. Mewada, "Identity Retention of Multiple Objects under Extreme Occlusion Scenarios using Feature Descriptors", *Journal of Communications Software and Systems*, vol. 14, no. 4, 2018, 290–301 DOI: 10.24138/jcomss.v14i4.541.

[4] H. K. Mewada, A. V. Patel, K. K. Mahant, "Concurrent design of active contour for image segmentation using Zynq ZC702", *Computers & Electrical Engineering*, vol. 72, 2018, 631–643 DOI: 10.1016/j.compeleceng.2018.01.024.

[5] W. Niu, J. Long, D. Han, Y.-F. Wang, "Human activity detection and recognition for video surveillance". In: *2004 IEEE International Conference on Multimedia and Expo (ICME)*, vol. 1, 2004, 719–722 DOI: 10.1109/ICME.2004.1394293.

[6] O. Javed, M. Shah, "Tracking and Object Classification for Automated Surveillance". In: *Computer Vision — ECCV 2002*, 2002, 343–357 DOI: 10.1007/3-540-47979-1_23.

[7] R. Diaz, J. Manuel, *Model-based object tracking with an infrared stereo camera*, Master Thesis, Örebro University, School of Science and Technology, 2015.

[8] A. Bolotnikova, *Articulated Object Tracking from Visual Sensory Data for Robotic Manipulation* Master Thesis, University of Tartu, Faculty of Science and Technology, Institute of Computer Science, 2017.

[9] K. L. Besseghieur, R. Trębiński, W. Kaczmarek, J. Panasiuk, "Trajectory tracking control for a nonholonomic mobile robot under ROS", *Journal of Physics: Conference Series*, vol. 1016, 2018 DOI: 10.1088/1742-6596/1016/1/012008.

[10] Y. Wei, Z. Lin, "Vision-based Tracking by a Quadrotor on ROS", *IFAC-PapersOnLine*, vol. 50, no. 1, 2017, 11447–11452 DOI: 10.1016/j.ifacol.2017.08.1814.

[11] G. Priyandoko, C. K. Wei, M. S. H. Achmad, "Human Following on ROS Framework a Mobile Robot", *SINERGI*, vol. 22, no. 2, 2018, 77–82 DOI: 10.22441/sinergi.2018.2.002.

[12] R. Collins, Y. Liu, M. Leordeanu, "Online selection of discriminative tracking features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, 2005, 1631–1643 DOI: 10.1109/TPAMI.2005.205.

[13] O. Javed, M. Shah, "Tracking and Object Classification for Automated Surveillance". In: *European Conference on Computer Vision — ECCV 2002*, 2002, 343–357 DOI: 10.1007/3-540-47979-1_23.

[14] B. Y. Lee, L. H. Liew, W. S. Cheah, Y. C. Wang, "Occlusion Handling in Videos Object Tracking: A Survey", *IOP Conference Series: Earth and Environmental Science*, vol. 18, 2014 DOI: 10.1088/1755-1315/18/1/012097.

[15] "PETS 2009 Benchmark Data". Binghamton University – State University of New York, http://cs.binghamton.edu/~mrldata/pets2009.html. Accessed on: 2020-02-20.

[16] B. Benfold, I. Reid, "Stable multi-target tracking in real-time surveillance video". In: *Computer Vision Pattern Recognition CVPR 2011*, 2011, 3457–3464 DOI: 10.1109/CVPR.2011.5995667.

[17] "Expectation Maximization algorithm to obtain Gaussian mixture models for ROS". Robotics with ROS, https://ros-developer.com/2017/11/16/expectation-maximization-algorithm-obtain-gaussian-mixture-models-ros. Accessed on: 2020-02-20.