

Sparse Representation Based Multi Object Tracking using GPU

Anuja Kumar Acharya, Rajalakshmi Satapathy, Biswajit Sahoo

Abstract: This work proposes a sparse based representation for tracking multi object for the longer sequence of video frame. Object of interest are first identified and then represented with set of low dimensional feature covariance matrix. These feature of different object are kept in a dictionary. In order to classify the object, sparse based Orthogonal matching pursuit(OMP) algorithm is used. Furthermore, towards reducing the computational overhead, proposed model is implemented on a graphical processing unit enhanced with the multi threaded resource for parallelization of the task. Experimental results shows that proposed method out perform as compared with the state of art in identifying the objects.

Keywords: Sparse representation , OMP , Feature Space , GPU, CUDA.

I. INTRODUCTION

Multi object tracking of non-rigid object for the longer sequence of video frame is a challenging task in the field of computer vision. Compare to single object tracking , Multi object tracking involve with larger computation and also required proper optimized code to search the candidate object within the time frame of the video sequence. GPU platform is the robust one in order to handle these computational complexity and memory usages of tracking. It is also important to focus that, accuracy is another key factor towards detecting the non rigid objects as these objects are generally undergoes deformation in shape and size. In the recent past, many appearance based models were developed to capture the pose and shape variation for single and multi object tracking[1-3]. In the literature, different methods are discussed towards handling the appearance variation that arises with real time tracking[4-5]. Although these methods are capable for detecting the object for smaller sequence but for longer sequence the accuracy of these method degrades. In the recent past, many researchers contribute articles towards detecting the single object using sparse optimization technique[6-10]. These sparse method use minimal reconstruction error in determining the tracking location of candidate object. Unfortunately, these methods have not utilize the sparsity concept for multi object. In the present work, we have developed a sparse based classification algorithm towards providing the accuracy of detecting or recognizing the objects.

Revised Manuscript Received on December 15, 2019.

Anuja Kumar Acharya*, School of Computer Engineering, KIIT University, Bhubaneswar , India. Email : anujafcs@kiit.ac.in .

Rajalakshmi Satapathy, School of Computer Engineering, KIIT Biswajit Sahoo, School of Computer Engineering, KIIT University, Bhubaneswar , India. Email : bsahoofcs@kiit.ac.in .

Furthermore in order to reduce the overload on the processor, proposed model is implemented using a multi threaded approach on a graphical processing Unit.

The paper is organized as follows. In Section II, we present the related work on single and multi object tracking. Section III, focus on the object detection and representation. Followed to this, Section IV, describes sparse representation based classification(SRC).Section V, describes the parallel Implementation of the proposed Model. In section 6, experimental results are described and finally, a conclusion closes this paper.

II. RELATED WORK

In this section we briefly discuss various issues related to single and multi object tracking especially appearance based method and sparse based method. Subspace based appearance model [11-13] uses the low dimensional covariance matrix of the image feature for finding the object in a frame. Skocaj and Leonardis [14], develop a subspace based learning method based on the weighted incremental PCA algorithm. Ross *et al.* [15], proposed an online incremental vision tracker(IVT), adopted with learning the online model with the appearance variation of the object. This method inhibits to detect the object under occlusion environment. Fragment-based tracker [17], a variation to the IVT tracker, was proposed to handle occlusions by using a patch-based appearance model. Although the use of subspace based model increase the reliability and robustness of tracking [10] but still it lacks in establishing the direct local relationship of pixel present in the image region. Tracking using covariance region descriptor is another frame work for detecting the object of interest in a video. Intern-sic and extern-sic parameter of the object is first captured and represented with low dimensional feature descriptors. F. Porikli, O.Tuzel[11] developed an appearance based covariance frame work for tracking object by capturing both the statistical and spatial attribute of the object window.

In the recent past, many research work are carried out on sparse represent based method for tracking. Mei.et al[16] developed sparse based method using particle filtering and template matching for classifying the objects. Weng et.al[17] create a dictionary to store the visual appearance of different object for uniquely distinguishing the objects. Many development are carried out in the recent past using GPU. Abramov et.al[18] used a mobile portable graphical processing unit(GPU) for a real time image segmentation to recognize the objects in the scene of the video. Chiang et al. [19] implemented a fast real-time content aware video targeting system.

Towards optimizing the target tracking, OMP algorithms are implemented on GPU environment. Feng et al. [20], use a parallel implementation for OMP algorithm towards use the matrix inversion and matrix-vector multiplication for uniquely classifying object to a class.

In this paper, we developed a two stage model frame work to track the object. In the 1st stage, object is represented using the covariance region descriptor, and then using sparse method classify the object. Followed to it, In the 2nd stage using the GPU and parallel method, proposed approach is implemented.

III. OBJECT DETECTION AND REPRESENTATION

Moving object can be identified in the target frame by background subtraction method [36] along with image filtering operation. Each of the detected object is represented with the bounding box. Using Tuzel et al. [8] covariance based representation, pixel present in the bounding box is convert into the feature matrix. i.e bounding box for the image regions are converted into the feature matrix using equation(1)

$$f_k = \begin{bmatrix} x & y & I(x,y) & I_x & I_y & \sqrt{I_x^2 + I_y^2} & \arctan \frac{I_x}{I_y} \end{bmatrix} \quad (1)$$

Here, x and y represent the coordinate position of the pixel in x and y direction respectively. $I(x, y)$ is the intensity value of the pixel I_x and I_y , is the derivative of the pixel in x and y direction respectively. As shown in the Figure1(f-h), Each object present in the scene of size $M \times N$ is converted into the feature matrix F_t of size $M \times N \times d$. This feature matrix F_t is further transferred into a low dimensional covariance matrix of dimension $d \times d$ and then rewritten into a vector a_{t_j} by reading the elements of the covariance matrix column wise.

$$COV(F_t) = \begin{pmatrix} c_{t,11} & c_{t,12} \dots & c_{t,1d} \\ c_{t,21} & c_{t,22} \dots & c_{t,2d} \\ \dots & \dots & \dots \\ c_{t,m1} & c_{t,m2} \dots & c_{t,md} \end{pmatrix}$$

$$\begin{bmatrix} C_{t,11} \\ C_{t,12} \\ \vdots \\ C_{t,d1} \\ \vdots \\ C_{t,dd} \end{bmatrix} = \begin{bmatrix} C_t^1 \\ C_t^2 \\ \vdots \\ C_t^{d+1} \\ \vdots \\ C_t^d \end{bmatrix} = a_t \dots \dots (2)$$

Finally, vector a_t of size $d^2 \times 1$ represent the object present in the scene.

IV. SPARSE REPRESENTATION BASED CLASSIFICATION

Sparse approximation is one of the popular method employed for classification problem. This method is used for classifying the object with the reduced cost and low memory. This algorithm can be expressed as followed

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_m \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} \dots & a_{1n} \\ a_{21} & a_{22} \dots & a_{2n} \\ \dots & \dots & \dots \\ a_{m1} & a_{m2} \dots & a_{mn} \end{pmatrix} X \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} \dots \dots (2)$$

In sparse based classification, we have provided with input test sample $y \in R^N$ and the dictionary $A \in R^{M \times N}$ and need to find a sparse vector X that gives the approximation solution $Y \cong A X$.

In our multi object sparse classification model, we have consider $y \in R^{d^2}$ be the test sample and $A \in R^{d^2 \times N}$ is the dictionary that keep all the object descriptors that is collected from the previous frame. Here d^2 represent the size of the feature covariance descriptor for an object and N represent the total number of object descriptor, present in the dictionary.

This N number of descriptors is again represented with the c number of class label i.e $A = [A_1 A_2 \dots A_c]$ and each object class $A = [a_{i1}, a_{i2} \dots a_{ik_i}] \in R^{d^2 \times k_i}$ where k_i is the number of samples belongs to class i and $N = \sum_{i=1}^c k_i$ is the total number of object descriptors present in the dictionary. So an element $a_{ij} \in R^{d^2}$ from the dictionary A represent the j^{th} training sample of class i .

In order to find the true label of an object, Y must be expressed in terms linear combination of A and X , thus

Y can be written as

$$Y = AX$$

Where $X \in (x_{11}, x_{12} \dots x_{ik_1} \dots x_{ck_i})$ is the coefficient vector corresponding to A

$$Y = [a_{11}, a_{12}, \dots a_{ck_i}] [x_{11}, x_{12}, \dots, x_{ck_i}]^T \dots 3)$$

This equation can be approximate as

$$y = [a_{i1}, a_{i2}, \dots a_{ik_i}] [x_{i1}, x_{i2}, \dots, x_{ik_i}]^T$$

Further it is written as

$$Y = A_i X_i$$

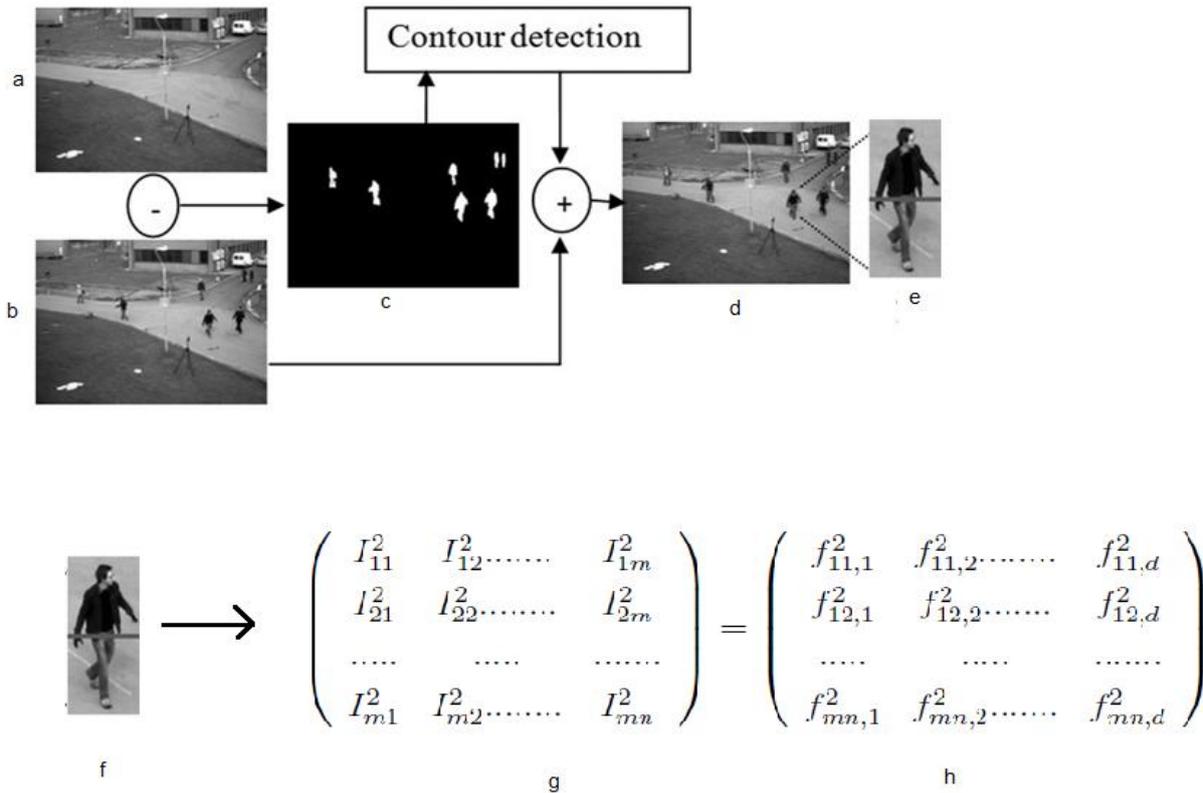


Figure 1: Proposed scheme for object tracking, (a)-(e) shows the background subtraction for detection of objects, (f)-(h) shows the representation of object with covariance descriptor

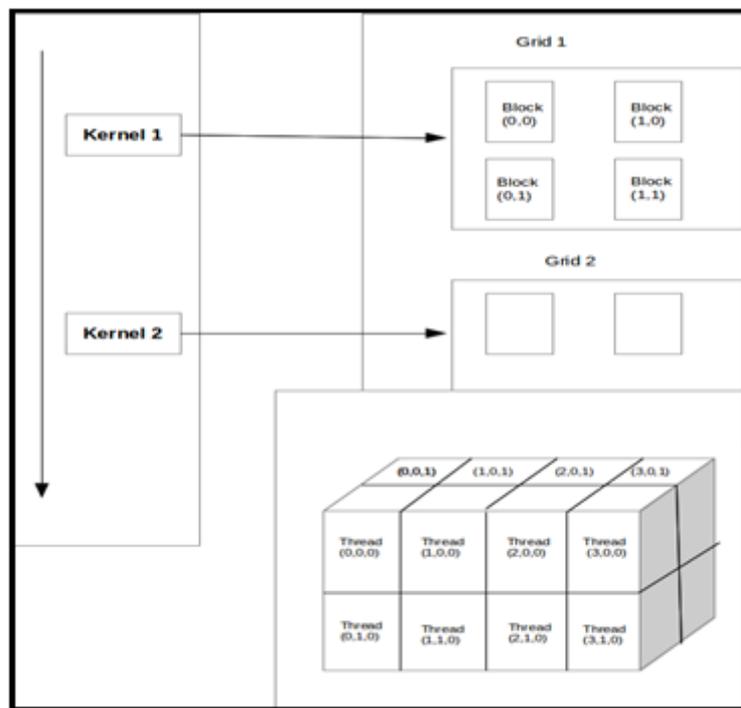


Figure 2: CUDA programming Model

Here, an object is classified into the i^{th} class subject to maximum or all the obtained nonzero coefficients of x are regrouped into the i^{th} class. Ideally, this can be stated as that the entries of x are all zero except those related to the same class as the test sample.

$$i.e Y = AX$$

where $X = [0, 0, \dots, x_{i1}, x_{i2}, \dots, x_{in_i} \dots \dots \dots]$

The residual $r_i = \|y - A_i(\hat{x})\|_2$ where x_i denotes the entries of x associated with the training samples from the i^{th} class. Finally, x is assigned a class label i that resulted in the minimal residual so that sparse set solution of x can be obtained by solving the

$$\hat{x} = \text{argmin} |\hat{x}|$$

We have used Orthogonal matching Pursuit(OMP) to find the solution of the above sparse problem. It is one of the greedy method and usually preferred for its speedy conversion rate. Here, in each iteration a local optimum solution is found. This is carried out by finding a column vector in A which is very close to the residual vector r . Initially the residual vector is assigned to the test vector Y . In each iteration, the residual vector is reassigned on the basis of evaluating $\|r_k - a_j x\|_2$.

In each iteration, one column vector is also selected on the basis of the minimum residual and then this selected column vector is removed for the next iteration. This process is repeated until the residual estimation reaches the maximum number of authorized iteration. As shown in the algorithm [1], OMP provide the solution to the approximation classification problem but it involve with the lots of computational task. This motivate us to develop the parallelization method for the OMP.

Algorithms 1 OMP algorithms

Input y and A

Output : x^t

Initialise $r_0 = y$, and $c_0 = \{\}$.

For $k \geq 1$ to t do

compute the residue $r_k = y - Ax$

Select a column A_j such that

$$j^* = \text{argmin} \|r_k - a_j x\|_2$$

Add $c_k = c_{k-1} \cup \{j^*\}$.

Coefficient Update : update the coefficient of (\hat{x}^k)

If $\|r_t\|_2 < \epsilon$ or t reaches the maximum number of authorized iteration the stop the algorithm.

end fors

V. PARALLEL IMPLEMENTATION OF OMP ON GPU

Identifying the group of independent instruction from the program is the first step of parallelism. In OMP algorithm, inner product computation is a single operation which is executed for n number of time for classifying the objects and is independent with respect to number of objects. Therefore, inner product has been parallelised using multiple thread.

A. GPU Execution Model

GPUs have been designed to run single instruction multiple data with the help of parallel thread. This processing unit generally consists of bundles of cards that packed with the hundreds of processor cores. NVIDIA contribution CUDA provides the programmer the flexibility and ease of using the parallel programming. The key component of the GPU environment is the Kernel which is mainly responsible to executes the parallel computation. A kernel is operated as a grid of thread blocks and all threads share data memory space. As show in the Figure 2, kernel is consists of one dimensional or the two dimensional grid. Each grid consists used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

of group of thread where group of threads are executed concurrently. All thread in a block execute the same program. Our parallel Orthogonal matching Pursuit (OMP) algorithm 2 for the multi object tracking is described as follows : We have considered p number of detected object for the classification. Input to our algorithm are y^p and A and the output is the coefficient vector x_i^{tl} . Here y_p denotes p number of detected objects and $A \in R^{d^2 \times N}$ is the dictionary that holds all the training samples. x_i^{tl} and r_i^{tl} represents the estimation of coefficient vector and the residual for object l at t^{th} iteration respectively. The processes of classification runs for t^{th} number of times for classifying an object and in each iteration we need to find a column that give maximum similarity or the minimum residual for the test object y_l . This column will be eliminated for the next iteration and it's index is added to the sparse vector x_l . For different object different number of iteration are required for reaching the termination condition. Finally, after all the iteration are over for an object, then we classify the object on the basis of the retrieving the sparse coefficient of the vector x_l^t .

Algorithms 2 Parallel OMP algorithm for Tracking

Input y^p and A

output : x_l^{tl}

Initialise $Q = 1, 2 \dots p, r_0^{tl} = y_l$, and $c_0 = \{\}$.

Select the Column A_i that solve maximization problem.

for each

$l \in Q$ $c_{tl}^l =$

$c_{t-1}^l \cup$

$\{\text{argmax}_{j=1..N} \|r_{tl}^l - a_j x_{tl}\|_2\}$.



if $\|r_{tl}^l\|_2 < \epsilon$ or t_l reaches the maximum number of authorized iteration

then $Q = Q - l$, otherwise $t_l = t_l + 1$.

end for

for each t_l update residue

$$r_{tl} = y_l - A\hat{x}_{tl}$$

Coefficient Update : Update the coefficient of (\hat{x}_{tl}^k)

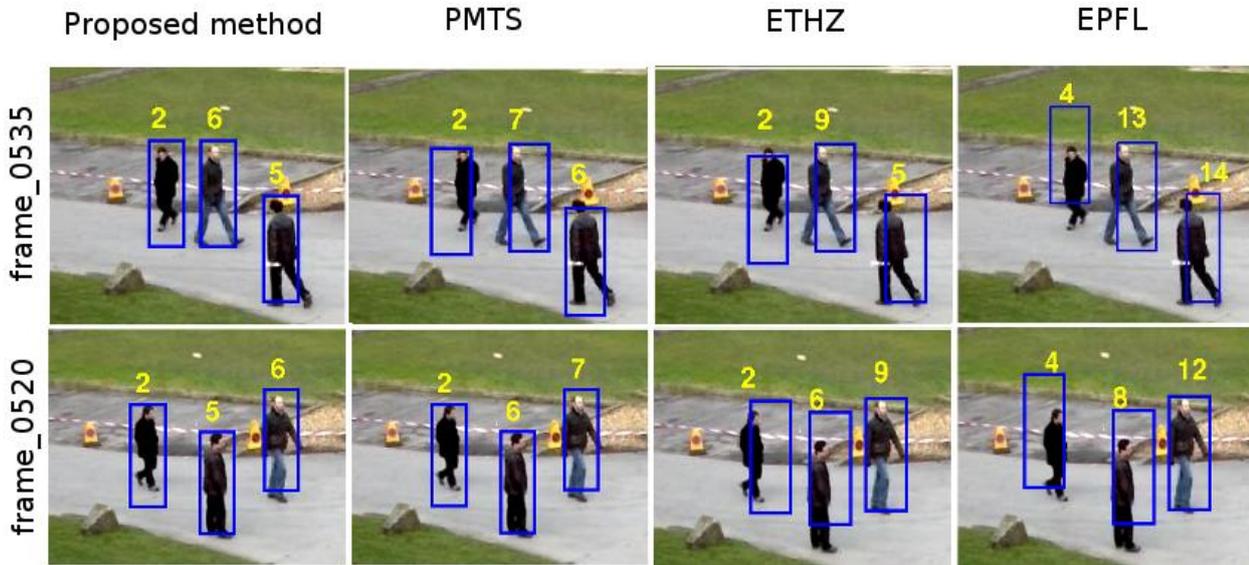


Figure 4: Comparison of our proposed method with PMTS, ETHZ and EPFL



Figure 5: Visual tracking result

For implementing the above algorithm in NVIDIA's GPU, we have launched the GPU kernel with two dimensional grid structures. 2D grid structure provide adequate number of blocks for executing parallel thread. In our model, we have assigned each block to hold the dictionary along with a test object y_l and run N number of parallel thread. N can be assigned with maximum value of 1024. Each block assigned with N parallel thread is used to estimate the inner product of the test sample y_l with the element of the dictionary. After t^{th} number of iteration each block produce the coefficient vector x_l for classifying the object y_l .

VI. RESULTS AND DISCUSSIONS

Our proposed algorithm is implemented in CUDA, and the program executed on a laptop supported with an Intel Core (TM) i7 CPU and 4 GB RAM. Furthermore the system is equipped with an NVIDIA GeForce GT 630M GPU card. Usually these cards are provided with two multiprocessors and 96 CUDA cores with 1 GB of global memory. Proposed method is applied on the PET's 09 standard dataset. We

compared our proposed method with the existing methods of the literature ETHZ [21], EPFL [22] and PMTS [23].

Object tracking Evaluation: We applied our proposed parallel implementation on the PET'09 database. Visual result of this implementation is shown in Figure 5. It is noted that objects are detected for the longer sequence of 400 frames without any false detection. In Figure 4, we compared the detection of different objects in the frame for our proposed method with ETHZ [21], EPFL [22] and PMTS [23].

Table I: MOTP and MOTA evaluation results.

N	MOTP	MOTA
ETHZ	54	52
EPFL	48	45
PMTS	68	72

Ours	72	78
------	----	----

Performance Measurement Metrics: In addition to the visual comparison, multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA) are two performance metrics that are used to compare our proposed method with the existing one. We obtained 72% of precision (MOTP) and 78% of accuracy (MOTA). These evaluation results show the robustness and accuracy of our algorithm.

Table 2: Comparison of Processing time in CPU and GPU for 5 person

Dictionary Size	GPU(ms)	CPU(ms)	Ratio(CPU/GPU) (ms)
100	2.6	53	0.25
200	2.13	8	3.75
300	2.33	12	5.15
400	2.56	14	5.4
500	2.7	17	6.29
600	2.9	20	6.89
700	3.1	25	8.06
800	3.2	27	8.4
900	3.3	29	8.78
1000	3.4	33	9.7

Execution Time

The complexity for computing the inner product is $O(M N d)$ and using [24] this complexity is approximated to $O(\log M N d)$. Execution time of the implemented OMP algorithm is evaluated and are compared for both GPU and CPU. Execution time of classification is estimated for five persons and shown in Table 2. Speed up rate of the GPU over the CPU are also shown in the Figure 6, From the figure it has been observed that GPU is computationally efficient as compared to the CPU and also processing time of GPU significantly reduced.

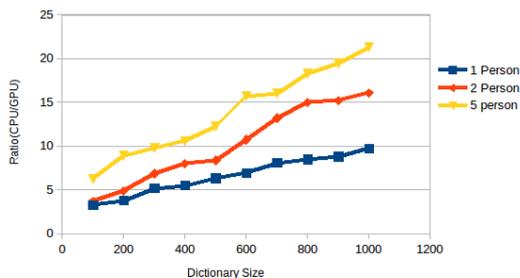


Figure 6: Speedup rate of GPU compare with CPU

VII. CONCLUSION

This paper addresses a sparse based classification technique for tracking multi object in real time video. OMP is used along with sparse technique to achieve the approximate solution to the multi class problem. objects are represented with covariance descriptor. Use of covariance descriptor increases the classification accuracy for different kind of objects. Furthermore, Our proposed algorithm is implemented with parallel thread on a graphical processing unit. This reduces considerably the processing time of algorithm for the object classification. Proposed algorithm shows the promising solution to ensure the detection of the moving object.

REFERENCES:

1. M. J Black, A.D Jepson, "Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation," *Int. J. Comput. Vision*,26(1) 63–84. doi:10.1023/A :1007939232436. URL <http://dx.doi.org/10.1023/A :1007939232436>,1998.
2. K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolution networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, Apr. 2016.
3. J. Matas, Z. Kalal, and K. Mikolajczyk, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
4. F. Huo, E.A Hendriks, "Multiple people tracking and pose estimation with occlusion estimation", *Computer Vision and Image Understanding*, 2012, 116, pp. 634–647
5. M. Andriluka, S. Roth and B. Schiel "People tracking by detection and People detection by tracking". *IEEE Conf. on CVPR*, 2008, pp. 1–8.
6. H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *CVPR*, 2011.
7. X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *ICCV*, 2009.
8. Q. Wang, F. Chen, W. Xu, and M.H. Yang, "Online discriminative object tracking with local sparse representation," in *WACV*, 2012.
9. X. Jia, H. Lu, and M.H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *CVPR*, 2012.
10. M. D. Breitenstein, F.Reichlin, B. Leibe, E. Koller Meier, and L. V. Gool, "Online multi-person tracking by- detection from a single and uncalibrated camera," *IEEE Trans. PAMI*, vol. 33, no. 9, pp. 1820–1833, 2011.
11. O.Tuzel, F.Porikli, P. Mee , " Covariance tracking using model update based on lie algebra,"in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1, pp. 728–735,2006.
12. Y. Li," On incremental and robust subspace learning,"*Pattern ecognition* ,37 ,1509-1518,2004.
13. K. Zhang, Q. Liu, Y. Wu, and M.-H. Yang, "Robust visual tracking via convolutional networks without training," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, Apr. 2016.
14. D.Skocaj, A.Leonardis, "Weighted and robust incremental method for subspace learning,"*Ninth IEEE International Conference on computer vision 2* , pg 1494-1501,2003.
15. J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
16. X. Mei, H. Ling, "Robust visual tracking and vehicle classification via sparse representation", *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011, 33, (11), pp. 2259–2272.
17. Q.Wang, F. Chen, W. Xu, "Online discriminative object tracking with local sparse representation". *IEEE Workshop on Applications of Computer Vision (WACV)*, January 2012, pp. 425–432.
18. A. Abramov, K. Pauwels, J. Papon, F. Worg and B. Dellen, Real-time segmentation of stereo videos on a portable system with a mobile GPU," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 22, no. 9, pp. 1292{1305, 2012.
19. C.-K. Chiang, S.-F. Wang, Y.-L. Chen, and S.-H. Lai, Fast JND-based video carving with gpu acceleration for real-time video retargeting., *IEEE Trans Circuits Syst. Video Techn.*, vol. 19, no. 11, pp. 1588{1597, 2009.

20. Y. Fang, J. Wu, L. Chen, and B. Huang, "Gpu implementation of orthogonal matching pursuit for compressive sensing," in *Proceedings of the 2011 IEEE 17th International Conference on Parallel and Distributed Systems, ICPADS '11*, (Washington, DC, USA), pp. 1044-1047, IEEE Computer Society, 2011.
21. M. D. Breitenstein, F. Reichlin, B. Leibe, E. Sudderth, and L. Van Gool, "Online multiperson tracking-by-detection from a single un-calibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 1820-1833, Sept. 2011.
22. F. Fleuret, P. Fua, H. Ben Shitrit, J. Berclaz, "Tracking multiple-people under global appearance constraints," in *Computer Vision (ICCV)*, 2011 IEEE International Conference on, 2011.
23. M. Elbahri, N. Taleb, K. Kpalma, and J. Ronsin, "parallel-algorithm implementation for multi-object tracking and surveillance.," *IET Computer Vision*, vol. 10, no. 3, pp. 202-211, 2016.
24. Y. Dai, D. He, Y. Fang, L. Yang, "Accelerating 2D orthogonal matching pursuit algorithm on GPU", *Journal of Super computing*, 2014, 69, (3), pp. 1363-1381.

AUTHORS PROFILE



Anuja Kumar Acharya, received his PhD in Computer Science & Engineering from KIIT University in 2017. He is currently working as an Asst. Prof in the School of Computer Engineering, KIIT University, India. His current research interest is in Image and parallel processing.



Rajalakshmi Satapathy, received her M.Tech from the Utkal University. She is currently working as Teaching Asst. in the School of Computer Engineering, KIIT University. Her research interest is in Data Science and Image processing.



Biswajit Sahoo, received his PhD from Utkal University. He is currently working as Professor in the School of Computer Engineering, KIIT University, India. His research interest is in Bioinformatics and Parallel Processing.