Context Driven MEC Resource Allocation for Time-Critical AR Applications

Savvas I. Raptis, Iason Karakostas, Nikolaos Dimitriou and Dimitrios Tzovaras
Information Technologies Institute
Centre for Research and Technology Hellas (CERTH)
Thessaloniki, Greece
savvasrapt@iti.gr, iason@iti.gr, nikdim@iti.gr, dimitrios.tzovaras@iti.gr

***Abstract*—Mobile Edge Computing (MEC) is a promising solution for delay-sensitive and computationally intensive applications mitigating the increased latency that cloud computing endures. This is achieved by delivering the computation resources in close proximity to the users' Mobile Devices (MDs) at the edge of the Radio Access Network (RAN). This network paradigm, has attracted the attention of researchers and several solutions for bandwidth and computation resources allocation have been proposed in order to achieve low latency and energy efficient models for offloading applications' tasks. In this paper we present an offloading optimization strategy, exploiting the visual information on each MD for allocating MEC resources. The significance of the several parameters in each video frame context is evaluated along with computation delay in a proper objective function. A game theory optimization algorithm is used for the objective function minimization leading to the optimal trade-off between latency and reliability.**

***Index Terms*—Augmented reality, cloud, edge, fog, computer vision, offloading optimization**

I. Introduction

The latest computer hardware advancements have allowed for development of Computer Vision (CV) methods able to perform real-time [1], [2]. However, the top performing methods [3] that offer the best results have extreme computational needs, rendering real-time execution on wearable devices impractical. At the same time, the progress in "wireless communications", such as the proliferation of 5G networks, have allowed for fast, low-latency communication of wearable devices with Cloud servers and Personal Computers. The combination of 5G capabilities of wearable/mobile devices and the computational power available in Edge/Cloud services allow real-time execution of computationally intensive AI models, offering results with improved quality.

Mobile Edge Computing (MEC) has been proposed as a solution for the high latency that conventional Mobile Cloud Computing (MCC) introduces, providing computation resources at the edge of Radio Access Network (RAN) nearby to the users' mobile devices [4]–[10]. MEC network performance has been a main subject of research towards achieving low computation delay [11]–[14], energy efficiency [14]–[16] and low cost [17], [18]. Those issues are addressed by modeling the computation delay, cost and energy consumption for task execution and data transmission and formulating the problem as an optimization objective function. Such models have been widely applied in Augmented Reality (AR) applications [6]–[10], [19], [20], allowing the utilization of latency-sensitive and computational-intense CV methods. Vehicular AR [8], [20], vehicle tracking and traffic warnings [19], unmanned aerial vehicles [4], [21] and security applications [22], [23] have exploited MEC infrastructures, in order to achieve low latency and energy efficiency.

In [19] a MEC network for vehicle tracking is proposed, assisted by its virtual "twin" at the cloud. Users may select to offload their tasks at either the real or virtual edge, which reduces successfully the computation latency. Also, in [20], [21], game theory optimization schemes are proposed for latency reduction in vehicular and UAV applications with MEC utilization. Users in such applications may observe highly cluttered and dangerous scenes, where the exploitation of more robust CV methods, executed on edge, would increase the quality of the scene analysis results. In [8] a vehicular MEC framework is proposed, consisting of a multi-tier network of edge base stations (BSs). Emergency situations are addressed by the first tier edge BSs, in close proximity to vehicles and pedestrians, ensuring low latency. However, an AR application of MEC with more restrictions of MEC resources, i.e., total number of BSs and MDs' computation capabilities, would require an application oriented resource allocation policy to address those issues.

MEC resource allocation may be combined with several heavy and lightweight versions of CV methods [9], [10], [23] leading to the best possible trade-off between latency and object detection accuracy. In this paper a context driven resource allocation scheme for time-critical AR applications is proposed. The scheme utilizes a game theory optimization algorithm, focusing on distributing access between powerful MEC processors and lightweight MDs based on the visual context of users field of view. Simultaneously, the scheme aims to keep average computation delay for all users in low levels. The contribution of this paper is summarized as follows:

- Estimation of the total delay according to users' offloading strategy through an analytical system model.
- Formulation of the problem in the form of an optimization objective function which combines the importance of low computation delay and high accuracy for emergency scenes.

- Utilization of a game theory optimization algorithm, modified according to the problem specifications, to obtain the solution of the resource allocation problem.

The following of this paper is structured as follows: In Section II the studied use case for AR application in the security domain is presented. In Section III an analytical model of the total system is formed, based on the various system parameters. More specifically, the application's required tasks, the RAN model to obtain the uploading data rate from each user to the BS and the total computation delay of users' tasks depending on their offloading strategy. In Section IV the users' offloading decision problem is defined and a game theory optimization algorithm [24] for the solution of the problem is presented. The results of the algorithm are examined through a numerical analysis in section V. Finally, the conclusion of this work are presented in Secton VII.

II. Application Description and Network Architecture

The presented offload optimization scheme can be implemented on ecosystems that employ wearable devices and mobile edge computing layers. DARLENE [23], is an AR ecosystem for law enforcement officers aiming to increase their situational awareness, by analyzing the scene of operation with CV methods and visualizing the results. The employed CV methods include, instance segmentation, target tracking, 2D pose estimation and human activity recognition that have to produce real-time results. A brief overview of the DARLENE architecture is depicted in Fig. 1. The field users are equipped with wearable MDs, namely Wearable Edge Computing Nodes (WECN) that have AR visors, cameras, an embedded PC and 5G connectivity. DARLENE offers a private network that allows the communication of the WECN devices with cloud services, headquarters etc. This network, offers 5G connectivity, and is deployed close to the field of operation, for example in a nearby patrol car that is also equipped with powerful MEC processors, namely Patrol Car Edge Nodes (PCEN), that can communicate with the WECN devices via the private wireless network. Through the 5G networking, connection to headquarters is possible and a "Command and Control" application allows the leading commander to have an overview of the site and better orchestrate the operation.

A main feature of the DARLENE ecosystem is the ability to exploit more powerful computational resources for real-time scene analysis. The computational node of the WECN incorporates lightweight versions of the CV methods that can perform real-time up to certain limits set by the number of detected objects. More robust and computational heavy versions of these methods are available for the edge/PCEN and cloud layer. These methods can produce better quality results, as shown in Fig. 2, and are also able to analyse more cluttered scenes. The CV methods employed on PCEN and Cloud layers, surpass the WECN variants on security specific datasets [25], by more that 10% on their respective metrics as indicated in Table I.

TABLE I: Quantitative results of WECN computer vision algorithms vs PCEN/Cloud variants for the task of Instance Segmentation (IS) and Human Pose Estimation (HPE). Evaluation was carried out on security oriented datasets [25]. The mean Average Precision is reported for IS and Object Keypoint Similarity for HPE.

	Instance Segmentation		Human Pose Estimation
	Person	Firearm	
WECN	49.0%	17.7%	67.5%
PCEN/Cloud	61.1%	32.8%	76.5%

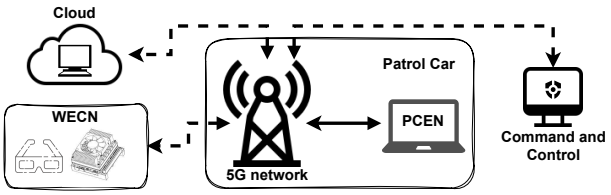
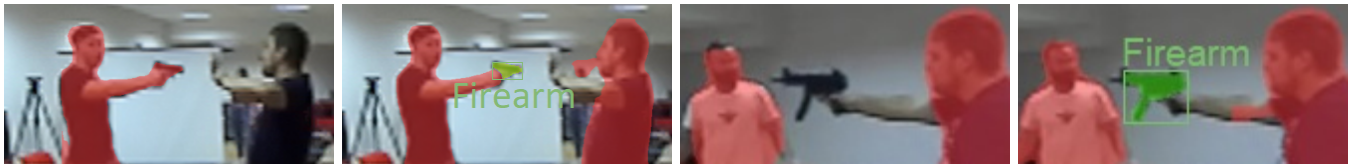


Fig. 1: Diagram of the DARLENE main components. Dashed lines indicate wireless connectivity and solid lines communication via cable.

The video captured by the cameras of the WECNs can be streamed to PCEN or Cloud computational nodes for CV analysis. Afterwards, the results will be transmitted to the wearable devices for visualization. However, as the number of users increases, it is not feasible to execute for each one scene analysis in real-time on a limited number of edge devices. To this end, the proposed optimization scheme allocates PCEN computational resources to DARLENE users, prioritizing complex scenes where security-related incidents are unfolding.

III. System Model

Consider a set of users $\mathcal{N} = \{1, \ldots, N\}$ equipped with a MD. Each user i is linked to video frame for processing, which begins with the dummy task START. The START task triggers the transmission of the video frame of data size d_{seg}, to the Instance Segmentation (IS) task $\{0\}$. The IS task aims to detect all humans and objects of interest (firearms, knives etc.) in the frame. For each located human, an image of data size d_{pos} is created defined by their respective bounding box, exploited by the Human Pose Estimation (HPE) task. The aim of HPE is to recognize a human's pose, thus it is applicable only for humans and not for the rest of the detected objects. HPE tasks are independent and can be executed at several devices simultaneously. Thus, if a number of K_i humans are recognized in the video frame associated with the i^{th} user, then a set of $\{1, \ldots, K_i\}$ HPE tasks have to be completed. When an HPE task is completed, the results data are transmitted to the dummy task END, denoting the ending of the video frame processing while the next frame can start. HPE task's result data is considered negligible compared to d_{seg} and d_{pos}. The application's procedure for the execution of the set of tasks $\mathcal{T}_i = \{0, 1, \ldots, K_i\}$ for user i are depicted through a Directed Acyclic Graph (DAG) [26] in Fig. 3. Each user can execute the application tasks locally or offload at the edge which consists of $\mathcal{M} = \{1, \ldots, M\}$ processors.



(a) Crop from video frame from analyzed on WECN. (b) Crop from video frame from analyzed on PCEN. (c) Crop from video frame from analyzed on WECN. (d) Crop from video frame from analyzed on PCEN.

Fig. 2: Qualitative comparison of WECN and PCEN layer computer vision methods results. In the second case, the method is able to produce better quality segmentation masks and also detect the firearms that is not detected at all in the WECN examples.

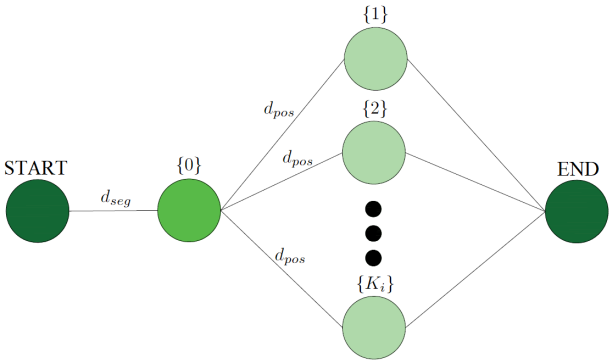


Fig. 3: Directed acyclic graph model of applications tasks.

A. Radio Access Network Model

The following model is an approximation the MDs' uploading data rates at some distance from the BS. In a real life scenario, this information is provided by the 5G RAN equipment (Amarisoft Callbox Mini) for every WECN. If r_i is the i^{th} user's distance from the BS, the received power from the MD is [27]:

$$P_{r,i}\,[dBm] = P_{t,i}\,[dBm] + 20\log(\frac{c}{4\pi f}) + \\ + 10\log(G_t) + 10\log(G_r) - 10n\log(r_i), \tag{1}$$

where r_i is expressed in metres, $P_{t,i}$ is the i-th user's MD transmitting power c is the speed of light, f is the central frequency of the transmitted signal, G_t and G_r are the transmitter and receiver antenna gains respectively and n is the path loss exponent, defined by the environment characteristics (e.g., urban, rural, industrial area, etc.). In order to define the i-th user's channel capacity C_i, the limitation of the maximum available capacity C_{max} should be taken into account:

$$C_i(r_i) = \begin{cases} C_{max}, & r_i < r_a \\ B\log_2(1 + \frac{P_{r,i}}{P_g}), & \text{otherwise} \end{cases}, \tag{2}$$

where B is the available bandwidth and P_g is the noise power and r_a is the maximum radius where C_{max} can be achieved.

B. Task Execution Delay Model

The processing time of a video frame for user i, is depended on all users' offloading strategies, i.e., the devices they choose to execute their tasks. The i-th user's offloading strategy is denoted with a vector $\mathbf{a}_i = [a_{i,1}\ a_{i,2}\ a_{i,3}]^T$. Every user may execute the IS task locally ($a_{i,1} = 0$) or offload them at a MEC processor ($a_{i,1} = 1$). In the first case, the HPE tasks are also executed at the selected MEC processor. Otherwise, part of the HPE tasks may also be offloaded, since those tasks can be executed in parallel. We denote with $a_{i,2} = 1, \ldots, M$ the user's choice for a MEC node to offload either the segmentation or part of the total pose estimation tasks. Finally, $a_{i,3}$ is the number of the HPE tasks that the user i choses to offload.

If the segmentation task execution time is t^l_{seg} and t^e_{seg} at the MD and a MEC node respectively, then an estimation for the total computation delay for this task is:

$$T_{i,seg} = (1 - a_{i,1})t^l_{seg} + a_{i,1}(t^e_{seg} + \frac{d_{seg}}{C_i} + t^q_{ij}), \tag{3}$$

where t^q_{ij} is the average queue waiting time at the node $j = a_{i,2}$ for user i. In the case that $a_{i,1} = 0$ (i.e., local execution of the IS task), the HPE tasks which are scheduled at the MEC node j are completed in:

$$T^e_{i,pos} = a_{i,3}(t^e_{pos} + \frac{d_{pos}}{C_i}) + t^q_{ij}, \tag{4}$$

where t^e_{pos} is the edge execution time at the edge. The HPE tasks scheduled for local execution is:

$$T^l_{i,pos} = (K_i - a_{i,3})t^l_{pos}, \tag{5}$$

where t^l_{pos} is the edge execution time at the MDs. Provided the i^{th} user's choice $\mathbf{a}_i$, the computation delay for the HPE tasks is:

$$T_{i,pos} = \begin{cases} K_i t^e_{pos}, & a_{i,1} \neq 0 \\ K_i t^l_{pos}, & a_{i,1} = a_{i,3} = 0 \\ \max([T^l_{i,pos}\ T^e_{i,pos}]), & \text{otherwise} \end{cases}, \tag{6}$$

and the total computation delay for video frame is:

$$T_i = T_{i,seg} + T_{i,pos}. \tag{7}$$

Finally, we compute the average queue waiting time. The maximum waiting time for user i at node j (i.e, the case that the user is last in the queue) is:

$$T^e_{ij} = \sum_{\substack{a_{n,2}=j \\ n\neq i}} a_{n,1}(t^e_{seg} + K_n t^e_{pos}) + (1 - a_{n,1})a_{n,3}t^e_{pos}. \tag{8}$$

The waiting time at node j is a random value, uniformly distributed between 0 and T^e_{ij}, that is

$$t^q_{ij} = \frac{1}{2}T^e_{ij}. \tag{9}$$

Algorithm 1 Human Pose Estimation Tasks Scheduling

$a_{i,3} \leftarrow 0$
$t_{d,MD} \leftarrow 0$
$t_{d,MEC} \leftarrow t_{ij}^{q}$
$k \leftarrow 0$
while $k < K_i$ **do**
  if $t_{d,MD} + t_{pos}^{l} < t_{d,MEC} + t_{pos}^{e} + \frac{\mathbf{d}_{pos}}{C_i}$ **then**
    $t_{d,MD} \leftarrow t_{d,MD} + t_{pos}^{l}$
  else
    $t_{d,MEC} \leftarrow t_{d,MEC} + t_{pos}^{e} + \frac{\mathbf{d}_{pos}}{C_i}$
    $a_{i,3} \leftarrow a_{i,3} + 1$
  end if
  $k \leftarrow k + 1$
end while

IV. Problem Formulation and Optimization Algorithm

A. Problem Formulation

MEC nodes provide the essential computational resources for more robust and heavy computational CV methods, compared to the lightweight versions of MDs. The computational resources of the system should be allocated, ensuring low average computational delay but, most importantly, high accuracy (in terms of instance segmentation and HPE results) to the users observing cluttered and/or dangerous scenes. The contextual content of users' video frames must be the major criterion for providing them with access to the MEC versions of the CV methods. If F_i are the detected weapons in i-th user's frame, the problem is the minimization of the objective:

$$L_i = \lambda_1(T_i - T_r) + (\lambda_2 K_i + \lambda_3 F_i)(1 - a_{i,1}), \tag{10}$$

where λ_1 and λ_2 and λ_3 are proper weights, notifying respectively the importance of the execution delay, HPE accuracy in a cluttered scene and HPE accuracy in weapon appearance observing scene. The problem formulation is expressed as:

$$\begin{aligned} \min_{\{\mathbf{a}_i\}} \quad & \frac{1}{N}\sum_{i=1}^{N} L_i, \\ \text{s.t.} \quad & a_{i,1} \in \{0,1\}, \quad a_{i,2} \in \mathcal{M}, \quad a_{i,3} \leq K_i \end{aligned} \tag{11}$$

B. Optimization Algorithm

For the minimization of the average value of the objective function, a game theory optimization algorithm is utilized (Algorithm 2). All users initiate their choice $\mathbf{a}_i$ according to (11). Considering that every user is aware of the other users' choices, they can compute the total load of every node from (8)-(9). The next step for every user is to obtain their optimal offloading strategy $\mathbf{a}_i'$ that minimizes L_i. In the case that $a_{i,1}' = 0$ and provided $a_{i,2}'$, $a_{i,3}'$ must be the number between 0 and K_i that minimizes $T_{i,pos}$ according to (8). This is achieved by executing a simple function described in Algorithm 1. Thus the problem reduces in obtaining the optimal $a_{i,1}'$ and $a_{i,2}'$, since $a_{i,3}'$ is calculated directly from

Algorithm 2 Game Theory Optimization for the Objective Minimization

$a_{i,1} \leftarrow$ random $0, 1$ for every $i \in \mathcal{N}$
$a_{i,2} \leftarrow$ random $j \in \mathcal{M}$ for every $i \in \mathcal{N}$
compute $a_{i,3}$ for every $i \in \mathcal{N}$ by Algorithm I
repeat
  for $i \in \mathcal{N}$ **do**
    $\mathbf{a}_i^{opt} \leftarrow \mathbf{a}_i$
    compute $L_i(\mathbf{a}_i^{opt})$ by (5)-(12)
    $L_{min} \leftarrow L_i(\mathbf{a}_i^{opt})$
    for $a_{i,1}' \leftarrow 0$ to 1 **do**
      for $a_{i,2}' \leftarrow 0$ to M **do**
        compute $a_{i,3}'$ by Algorithm 1
        compute $L_i(\mathbf{a}_i')$ by (5)-(12)
        if $L_i(\mathbf{a}_i') < L_{min}$ **then**
          $\mathbf{a}_i^{opt} \leftarrow \mathbf{a}_i'$
          $L_{min} \leftarrow L_i(\mathbf{a}_i')$
        end if
      end for
    end for
    if $a_{i,1}^{opt} = 0$ **then** $\mathbf{a}_i \leftarrow \mathbf{a}_i^{opt}$
  end for
  find set of users $\mathcal{N}' \in \mathcal{N}$ with $a_{i,1}^{opt} = 1$ and $a_{i,2}^{opt} \neq a_{i,2}$
  $p \leftarrow$ random $i \in \mathcal{N}'$
  $\mathbf{a}_p \leftarrow \mathbf{a}_p^{opt}$
until END message is received

Algorithm 1. Subsequently the users find the choice $\mathbf{a}_i'$ with the minimum L_i. If the optimal choice of user i is to execute the segmentation task at the mobile device, which is expressed as $a_{i,1}' = 0$, the system confirms the choice. In the case that user i has chosen to offload the segmentation task for execution at the current node ($a_{i,2}' = a_{i,2}$), the choice is also confirmed by the system. All users which obtain an alternative node ($a_{i,2}' \neq a_{i,2}$) to offload their segmentation tasks send a request message to the system and only one user's optimal choice is randomly confirmed. The algorithm repeats itself until no further optimization occurs. Fair game is guaranteed since all MDs execute identical algorithms with zero human input.

V. Numerical Results

In this Section the operation of the system is theoretically examined, using numerical calculations. A total number of 200 executions of Algorithm 2 are performed in order to calculate the average objective function value against the number of decision slots and the percentage of the users which offload their segmentation tasks according to their observing scenes.

The system's behavior when changes in the observing scenes occur (i.e., high clutter and weapon appearances) is also investigated. For such scenarios, the users' choices are initialized with the optimized results and the numerical analysis is repeated for a rearranged scenario. As a change of clutter in the observing scene, the users related with the maximum and minimum scene clutter are selected and their observing

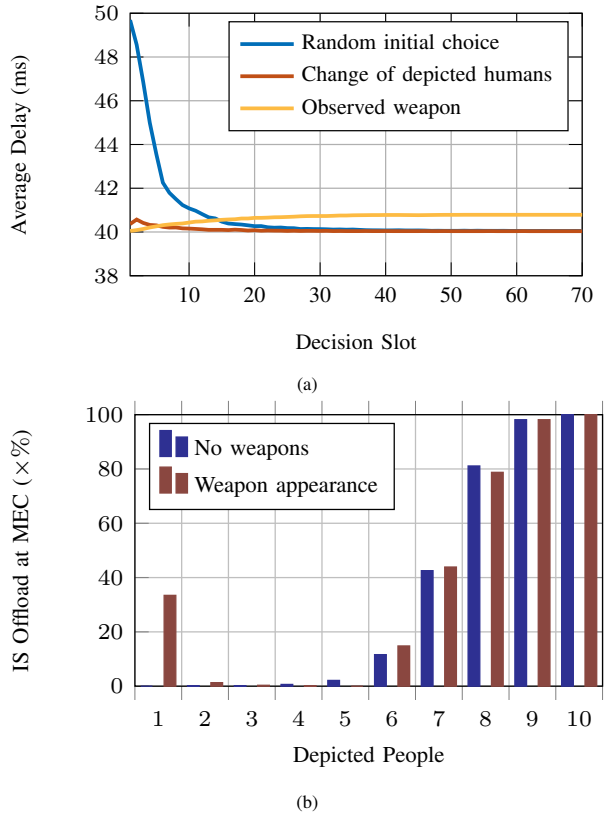


Fig. 4: (a) Average computation delay and (b) percentage of users offloading the segmentation task for equal users' uploading data rates.

scenes are reversed. For a weapon appearance scenario, an experiment is carried out where a weapon is detected by the MD of the user with the less scene clutter. This case can be considered as the most challenging since that user, otherwise, would be last in priority.

The system parameters are set as follows: The number of located humans K_i is random between 1 and 10. The segmentation task's execution times are $t^l_{seg} = 30$ ms and $t^e_{seg} = 20$ ms. The pose estimation tasks execution times are $t^l_{pos} = 2.5$ ms and $t^e_{pos} = 1.5$ ms. The tasks' required data are $d_{seg} = 7.4$ Mb and $d_{pos} = 1.2$ Mb. The number of users is $N = 30$ and the number of MEC processors is $M = 10$.

A. Equal Uploading Data Rate for all Users

In order to obtain the proper weights λ_1, λ_2 and λ_3 the numerical analysis is performed independently of users' uploading data rates setting for all users $C_i = 90$ Mbps. Setting $\lambda_2 = 1$ and considering no weapon appearance, sufficient results are obtained for $\lambda_1 = 2.5$. Similarly, the analysis with the selected λ_1 and λ_2 is performed in a weapon appearance scenario for several values of λ_3. The desired results are obtained for $\lambda_3 = 200$. For this weight selection the results of the numerical experiments are shown in Fig. 4. The proposed optimization algorithm chooses successfully the users with the highest located people to offload their segmentation tasks and minimizes the objective function in 40 decision slots. The algorithm also, sufficiently responses in the rearranged scenario in 20 desicion slots. For the scenario of weapon appearance, an increase in the average delay is observed, illustrated in Fig. 4 (a). This behavior is expected, since MEC computational resources are consumed by a user with a minor number of located people. However system keeps a low average users' delay maintaining real-time performance. The percentage of MDs which offload their tasks is shown in Fig. 4 (b). In this case, the algorithm chooses the users with the highest number of located people to offload their segmentation task. Also, at 100% of the experiments the MD with the observed weapon is prioritized to offload its segmentation task.

B. Diverse Uploading Data Rates for Users' MDs placed in Circular Cell

Since the proper weights are obtained in the previous numerical analysis, the more realistic case where the users are randomly placed in a circular cell is investigated. This results in a variance of their data uploading capability. A cell of range $R = 15$ m is examined. The uploading data rate of a user's MD is given by (2) and the maximum upload data rate is $C_{max} = 133$ Mbps. The transmit and receive antennas are considered isotropic, thus $G_t = G_r = 1$. The average frequency is $f = 5$ GHz and the speed of light is $c = 3 \times 10^8$ m/s. The available bandwidth for a user is $B = 20$ MHz and the transmission power is $P_{t,i} = 1$ mW. The noise power is considered $P_g = -100$ dBm and the loss coefficient of distance is $n = 4$. By those parameters the users' uploading data rate with respect to their position is obtained.

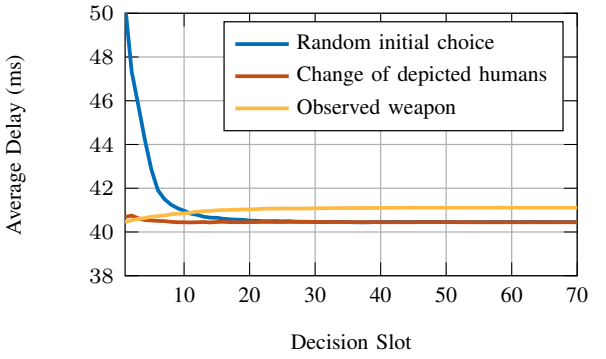
The same set of experiments with Section III.A are performed and the results are presented in Fig. 5. In this scenario the user related to a weapon depiction always chooses to offload the segmentation task at a MEC processor. The offloading percentage of users with highly cluttered scenes is decreased, which is expected since they may have low uploading rates, thus local execution of their segmentation tasks may give a better trade-off in latency and accuracy.

VI. Conclusions and Future Work

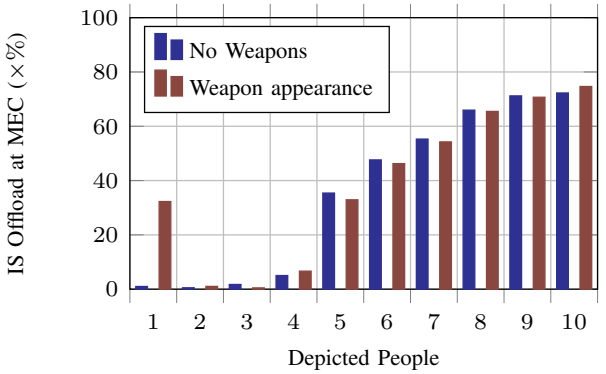
A context driven task offloading strategy has been presented, addressing the requirements of a time critical AR application for fast and highly reliable computation results for users in emerging situations. An optimization scheme is employed, towards better utilization of MEC recources in order to achieve the best possible trade-off between latency and object detection accuracy. MEC resource allocation for AR, based on video frame content is applicable in several applications which include emergency situations. Future work may include further examination of the problem including optimizing energy consumption, involving the MDs' battery level in the problem formulation, and real-life experiments, introducing virtual scenarios with several numbers of users and observing scenes.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme



(a)



(b)

Fig. 5: (a) Average computation delay and (b) percentage of users offloading the segmentation task for distributed MDs in a cell range.

under grant agreement No 883297 (project DARLENE). This publication reflects only the authors' views. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] I. Karakostas, V. Mygdalis, A. Tefas, and I. Pitas, "Occlusion detection and drift-avoidance framework for 2d visual object tracking," *Signal Processing: Image Communication*, vol. 90, p. 116011, 2021.

[2] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas, and I. Pitas, "Embedded uav real-time visual object detection and tracking," in *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 708–713, IEEE, 2019.

[3] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, *et al.*, "Internimage: Exploring large-scale vision foundation models with deformable convolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14408–14419, 2023.

[4] S. A. Huda and S. Moh, "Survey on computation offloading in UAV-enabled mobile edge computing," *Journal of Network and Computer Applications*, vol. 201, p. 103341, 2022.

[5] A. Islam, A. Debnath, M. Ghose, and S. Chakraborty, "A survey on task offloading in multi-access edge computing," *Journal of Systems Architecture*, vol. 118, p. 102225, 2021.

[6] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "A survey on mobile augmented reality with 5g mobile edge computing: Architectures, applications, and technical aspects," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1160–1192, 2021.

[7] D. G. Morín, P. Pérez, and A. G. Armada, "Toward the distributed implementation of immersive augmented reality architectures on 5g networks," *IEEE Communications Magazine*, vol. 60, no. 2, pp. 46–52, 2022.

[8] P. Zhou, W. Zhang, T. Braud, P. Hui, and J. Kangasharju, "Enhanced augmented reality applications in vehicle-to-edge networks," in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pp. 167–174, 2019.

[9] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, (New York, NY, USA), Association for Computing Machinery, 2019.

[10] Q. Liu and T. Han, "Dare: Dynamic adaptive mobile augmented reality with edge computing," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*, pp. 1–11, 2018.

[11] L. A. Haibeh, M. C. Yagoub, and A. Jarray, "A survey on mobile edge computing infrastructure: Design, resource management, and optimization approaches," *IEEE Access*, vol. 10, pp. 27591–27610, 2022.

[12] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via markov decision process in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2483–2493, 2021.

[13] L. Liu, B. Sun, Y. Wu, and D. H. K. Tsang, "Latency optimization for computation offloading with hybrid noma–oma transmission," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6677–6691, 2021.

[14] F. Sufyan and A. Banerjee, "Computation offloading for distributed mobile edge computing network: A multiobjective approach," *IEEE Access*, vol. 8, pp. 149915–149930, 2020.

[15] P. Padidem and A. Lee, "Studying offloading optimization for energy-latency tradeoff with collaborative edge computing," in *2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pp. 1–6, IEEE, 2022.

[16] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, "Joint computation offloading and scheduling optimization of iot applications in fog networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 3266–3278, 2020.

[17] D. T. Wojtowicz, S. Yin, F. Morvan, and A. Hameurlain, "Cost-effective dynamic optimisation for multi-cloud queries," in *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, pp. 387–397, 2021.

[18] A. Mehta and L. Eleftheriadis, "Smart edge power management to improve availability and cost-efficiency of edge cloud," in *2022 IEEE 15th International Conference on Cloud Computing (CLOUD)*, pp. 125–133, 2022.

[19] Y. Wang, S. Chen, Y. Xia, D. Melissourgos, and H. Wang, "Dynamic edge-twin computing for vehicle tracking," in *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, pp. 106–111, 2021.

[20] Y. Liu, S. Wang, J. Huang, and F. Yang, "A computation offloading algorithm based on game theory for vehicular edge networks," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2018.

[21] M.-A. Messous, S.-M. Senouci, H. Sedjelmaci, and S. Cherkaoui, "A game theory based efficient computation offloading in an uav network," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4964–4974, 2019.

[22] N. Dimitriou, G. Kioumourtzis, A. Sideris, G. Stavropoulos, E. Taka, N. Zotos, G. Leventakis, and D. Tzovaras, "An integrated framework for the timely detection of petty crimes," in *2017 European Intelligence and Security Informatics Conference (EISIC)*, pp. 24–31, 2017.

[23] K. C. Apostolakis, N. Dimitriou, G. Margetis, S. Ntoa, D. Tzovaras, and C. Stephanidis, "Darlene–improving situational awareness of european law enforcement agents through a combination of augmented reality and artificial intelligence solutions," *Open Research Europe*, vol. 1, p. 87, 2022.

[24] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[25] N. Kilis, G. Tsipouridis, I. Karakostas, N. Dimitriou, and D. Tzovaras, "Augmentation based on artificial occlusions for resilient instance segmentation," in *International Conference on Image Analysis and Processing, to be published*, 2023.

[26] S. Sundar and B. Liang, "Offloading dependent tasks with communication delay and deadline constraint," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 37–45, 2018.

[27] S. Sun, T. A. Thomas, T. S. Rappaport, H. Nguyen, I. Z. Kovacs, and I. Rodriguez, "Path loss, shadow fading, and line-of-sight probability models for 5g urban macro-cellular scenarios," in *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–7, 2015.