PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Edge-to-fog computing for color-assisted moving object detection

Ying Liu, Zachary Bellay, Payton Bradsky, Glen Chandler, Brandon Craig

Ying Liu, Zachary Bellay, Payton Bradsky, Glen Chandler, Brandon Craig, "Edge-to-fog computing for color-assisted moving object detection," Proc. SPIE 10989, Big Data: Learning, Analytics, and Applications, 1098903 (13 May 2019); doi: 10.1117/12.2516023

Event: SPIE Defense + Commercial Sensing, 2019, Baltimore, Maryland, United States

Edge-to-Fog Computing for Color-Assisted Moving Object Detection

Ying Liu, Zachary Bellay, Payton Bradsky, Glen Chandler, and Brandon Craig

Department of Computer Engineering, Santa Clara University
Santa Clara, CA 95053 USA

ABSTRACT

Future Internet-of-Things (IoT) will be featured by ubiquitous and pervasive vision sensors that generate enormous amount of streaming videos. The ability to analyze the big video data in a timely manner is essential to delay-sensitive applications, such as autonomous vehicles and body-worn cameras for police forces. Due to the limitation of computing power and storage capacity on local devices, the fog computing paradigm has been developed in recent years to process big sensor data closer to the end users while it avoids the transmission delay and huge uplink bandwidth requirements in cloud-based data analysis. In this work, we propose an edge-to-fog computing framework for object detection from surveillance videos. Videos are captured locally at an edge device and sent to fog nodes for color-assisted L_1-subspace background modeling. The results are then sent back to the edge device for data fusion and final object detection. Experimental studies demonstrate that the proposed color-assisted background modeling offers more diversity than pure luminance based background modeling and hence achieves higher object detection accuracy. Meanwhile, the proposed edge-to-fog paradigm leverages the computing resources on multiple platforms.

Keywords: Big data, color video, edge-to-fog computing, Internet-of-things, L_1-norm maximization, object detection, subspace learning, video surveillance.

1. INTRODUCTION

Detecting moving objects from streaming videos is an important task for visual surveillance systems. To tackle this problem, the usual approach is to model the background and perform background subtraction.[1–3] A classical background modeling scheme is through principal-component analysis (PCA).[4,5] Since the background scene of a surveillance video can be modeled as a low-rank component, PCA is able to find the low-rank subspace in which the background scene lies. In recent years, the L_1-norm PCA method[6–8] has been developed as a robust subspace learning method, and is successfully applied to the video background modeling problem.[9–12] When the captured video data have corruptions, L_1-PCA estimates more accurate background scenes compared to traditional L_2-PCA methods.

As surveillance cameras become ubiquitous in the Internet-of-Things (IoT),[13] the amount of video data increases drastically, and processing these big data requires more computing resources. While cloud computing offers abundant computation capability, it causes large transmission delay and requires huge uplink bandwidth. For delay-sensitive tasks such as autonomous vehicles and body-worn cameras for police forces, the ability to detect moving objects in a timely manner is essential. Hence, processing video data on the edge devices that capture these data is more preferable. The so-called edge computing paradigm thus becomes increasingly prevalent in IoT applications ranging from face recognition to object detection and human action recognition.[14] Nevertheless, an IoT edge device is often resource constrained and has limited processing power to run sophisticated computer vision algorithms. Therefore, it is inevitable to trade off computational resources, frame rate, and accuracy on this single edge device.

In contrast, fog computing seeks to address some of the issues that arise when using resource-constrained edge devices in delay-sensitive scenarios. By offloading much of the computationally complex processing to powerful, local machines, the system is able to take advantage of the high-accuracy provided by computationally complex algorithms while engaging the low-latency provided by keeping computation on or near the edge.[14] In essence, the edge device in an edge-to-fog network does little to none of the computation, while the data from

Big Data: Learning, Analytics, and Applications, edited by Fauzia Ahmad,
Proc. of SPIE Vol. 10989, 1098903 · © 2019 SPIE · CCC code:
0277-786X/19/$18 · doi: 10.1117/12.2516023

an edge device is transmitted to local servers referred to as "fog nodes" that take on most of the computational load. By reducing the amount of computation done by the edge device itself, the edge device is free to capture video at higher frame rate, while benefiting from more abundant resources offered by nearby fog devices. The fog computing paradigm has been utilized in several computer vision tasks. For example, fog computing based urban speeding traffic monitoring system that analyzes video streams captured by drones,[15] and face recognition systems using the mobile-cloudlet/fog-cloud architecture.[16,17] Nevertheless, these fog computing frameworks still have drawbacks, such as the limitation of a single fog node,[15,18] and the dependence on a remote cloud.[16,17]

In this work, we propose an edge-to-fog computing paradigm to detect moving objects with low latency. The edge device is able to capture images and split them into pieces that are transmitted to nearby servers for multi-core, parallel processing. Since these larger machines have more memory and processing power, they are able to receive images, process them, and send them back to the edge device in a timely manner. This edge-to-fog computing process is faster than utilizing a single edge device to process the images, giving greater power to IoT systems. On the other hand, because data communication is in the local area network, bottle-necking effects and latency of transmission to a cloud server are reduced.

Existing L_1-PCA method for background modeling is solely based on the luminance component of the video sequences.[9–12] To utilize the color information in the captured video sequence, we propose a pixel-wise L_1-subspace learning method in the red, green, and blue (RGB) color space. With the additional RGB components of each image frame as opposed to a single luminance component, we have improved prediction accuracy for moving object detection. Ultimately, we have optimized the speed and accuracy of moving-object detection to meet the increasing demands of future IoT devices.

The rest of the paper is organized as follows. In Section 2, we introduce the proposed color-assisted L_1-subspace learning algorithm for surveillance video background modeling. In Section 3, we introduce the proposed edge-to-fog computing paradigm. Section 4 presents experimental studies of our proposed system with a publicly available, offline data set as well as real-time camera data. Finally, we draw conclusions and outline future research opportunities in Section 5.

2. PROPOSED COLOR-ASSISTED L_1-SUBSPACE LEARNING

The theoretical foundation of the proposed video analysis system is the recent advances in L_1-norm principal-component analysis (L_1-PCA), or L_1-subspace learning. L_1-subspace learning is the procedure to calculate robust principal components of the underlying data. In conventional PCA, for a collection of N D-dimensional data samples $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$, a linear rank-$r$ subspace can be found by SVD of the data matrix $\mathbf{X}$, or equivalently, L_2-PCA of the following form

$$\mathbf{P}_{L_2} = \arg \max_{\substack{\mathbf{P} \in \mathbb{R}^{D \times r} \\ \mathbf{P}^T \mathbf{P} = \mathbf{I}_r}} \|\mathbf{X}^T \mathbf{P}\|_2, \tag{1}$$

and the resulting columns in $\mathbf{P}_{L_2}$ are named L_2 principal components. These L_2 principal components span a rank-r subspace in which the representation error of the data samples is minimized. However, in the presence of outliers in the data samples, the L_2-subspace can severely deviate from the true rank-r subspace of the nominal data, and the rank-r representation in this L_2-subspace cannot faithfully characterize the nominal data.

In contrast, L_1-PCA calculates $\mathbf{P}_{L_1} \in \mathbb{R}^{D \times r}$ via L_1-norm data projection maximization[6,7,19–21] in the following form, which pursues a robust subspace representation for the data matrix $\mathbf{X}$,

$$\mathcal{P}^{L_1} : \mathbf{P}_{L_1} = \arg \max_{\substack{\mathbf{P} \in \mathbb{R}^{D \times r} \\ \mathbf{P}^{\mathrm{T}} \mathbf{P} = \mathbf{I}_r}} \|\mathbf{X}^{\mathrm{T}} \mathbf{P}\|_1. \tag{2}$$

Since projecting all the data to a subspace deviated by the outliers is less likely to generate a larger projection L_1-norm than projecting the data to the correct low-rank subspace, the calculated L_1-subspace $\mathbf{P}_{L_1}$ is likely to be closer to the correct subspace than L_2-PCA calculated subspace. The r columns of $\mathbf{P}_{L_1}$ are the so-called r L_1 principal components that describe the rank-r subspace in which $\mathbf{X}$ lies.

In recent years, L_1-subspace learning has been successfully applied to computer vision tasks such as video surveillance and face recognition.[9–12,22,23] In particular, the L_1-subspace can be learned directly in compressed domain for moving objection detection,[9] and can also be updated in an online fashion for surveillance video background modeling.[10]

Nevertheless, the methods aforementioned convert color videos to gray-scale components and process only the gray-scale components for moving object detection. In this section, we propose to learn an L_1-subspace individually for each pixel in the original RGB color space to enhance the moving object detection accuracy.

Consider a color video sequence of T frames. Each frame has a resolution of $m \times n$ pixels. Let $\mathbf{x}_t^i = [r_t^i, g_t^i, b_t^i]^T$ denote the ith pixel of the tth video frame, consisted of the red r_t^i, green g_t^i and blue b_t^i values, each within the range $[0, 255]$ and thus has 8-bit representation. The intensity values of the ith pixel across T frames are collected and form a matrix

$$\mathbf{X}^i = [\mathbf{x}_1^i, \mathbf{x}_2^i, \cdots, \mathbf{x}_T^i] \in \mathbb{R}^{3\times T}. \tag{3}$$

Then the rank-1 L_1-PCA is applied to $\mathbf{X}^i$ to obtain the rank-1 L_1-subspace for the ith pixel

$$\mathbf{p}^i = \arg \max_{\substack{\mathbf{p}\in\mathbb{R}^3 \\ \|\mathbf{p}\|_2=1}} \|\mathbf{p}^T\mathbf{X}^i\|_1. \tag{4}$$

This procedure is repeated for all pixels in the video frame, hence resulting in D L_1-subspaces $\mathbf{p}^i$, $i = 1, 2, \cdots, D$, $D = m \times n$.

To detect the moving object, for the i-th pixel $\mathbf{x}_t^i$, we first calculate its L_2-distance to the corresponding RGB L_1-subspace $\mathbf{p}^i$:

$$d_t^i = \|\mathbf{x}_t^i - \mathbf{p}^i\mathbf{p}^{i\ T}\mathbf{x}_t^i\|_2, \tag{5}$$

then compare d_t^i to a predefined threshold η. If $d_t^i > \eta$, the pixel is classified as the moving object, otherwise, it is classified as the background. Hence a binary mask $\mathbf{M}_t^{\text{rgb}} \in \{0,1\}^{m\times n}$ indicating the moving object pixels and the background pixels can be formed for the tth frame, where the superscript "rgb" indicates the binary mask is generated by processing the RGB color frame.

Besides, we also process the gray-scale video frames using a simple mean filter to estimate the background. That is, the background of the ith pixel is estimated by the average of the luminance intensity values of that pixel over T frames by

$$b^i = \frac{1}{T}\sum_{t=1}^{T} y_t^i, \tag{6}$$

where y_t^i stands for the Y component (luminance component) of the ith pixel of the tth frame. For moving object detection, we calculate the absolution difference between b_i and y_t^i by

$$D_t^i = |y_t^i - b_i|. \tag{7}$$

If D_t^i is larger than a predefined threshold γ, then y_t^i is classified as the moving object, otherwise, it is classified as the background pixel. Hence, a second binary mask $\mathbf{M}_t^{\text{gray}} \in \{0,1\}^{m\times n}$ can be formed for the tth frame, where the superscript "gray" indicates the binary mask is generated by processing the gray-scale frame. Afterwards, we take the union of the two binary masks $\mathbf{M}_t^{\text{rgb}}$ and $\mathbf{M}_t^{\text{gray}}$ to determine the final binary mask $\mathbf{M}_t \in \{0,1\}^{m\times n}$ for the tth frame.

3. PROPOSED EDGE-TO-FOG COMPUTING PARADIGM

In order to achieve low-latency analysis of video data for the purpose of efficient and effective moving-object detection, the edge-to-fog computing paradigm is utilized in this research. As discussed in the introduction, a simple edge computing infrastructure alone is not sufficient to accommodate the low-latency requirements of this system. Because the device used to capture image data is resource constrained and has limited processing power to run essential computer vision algorithms, a system consisting exclusively of the edge device would be unable to process the camera data in time. Hence, a combination of edge and fog computing is proposed to achieve the combination of high-accuracy and low-latency that is prohibited by pure cloud computing or pure edge computing.

The proposed edge-to-fog system combines the speed of local network processing found in edge systems with the computational power of distributed computing by employing parallel processing and computational offloading within a local network. The infrastructure of this system is found in Figure 1.

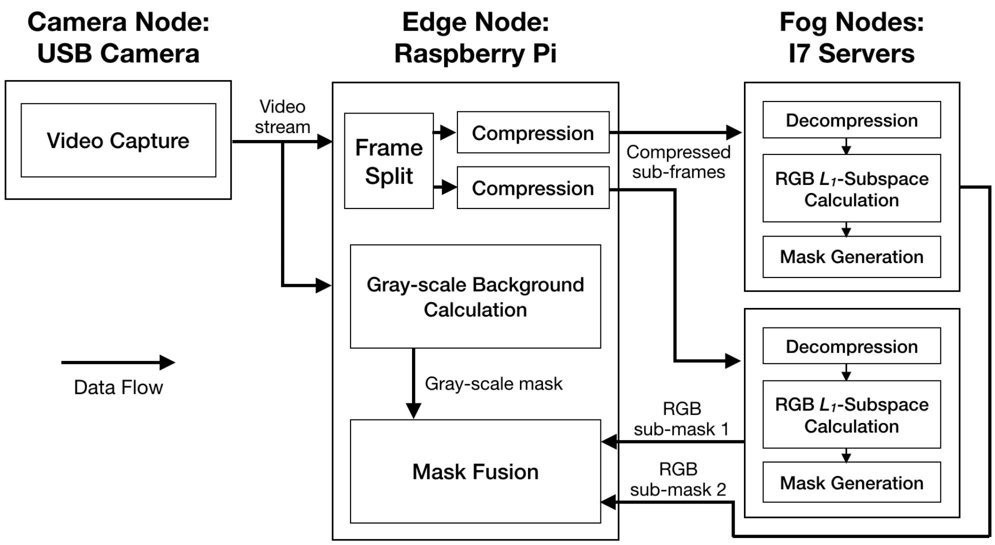


Figure 1: Edge-to-Fog Computing Architecture

The edge-to-fog network is based on a client-server architecture. The edge device streams video as the client that offloads the RGB L_1-subspace computation to the fog nodes. The edge device will capture the video stream, divide the image into two sub-images, send each sub-image to their respective fog nodes for L_1-subspace processing within the RGB color space, generate binary masks of the sub-images. Parallel to the offloaded processing of the RGB L_1-subspace, the edge device will process the gray-scale component of the same frame and create another object binary mask locally. The resulting binary mask of the sub-images from each fog node will then be sent back to the edge device and reconstructed into the full mask. The binary masks generated by processing the RGB frame and gray-scale frame will then be combined at the edge device, and the final detection of the the moving object will be displayed.

By offloading the RGB L_1-subspace computation to the fog nodes, the system will achieve low-latency video data processing. Meanwhile, moving-object detection accuracy is improved by taking advantage of the color-assisted L_1-subspace background modeling.

4. EXPERIMENTAL STUDIES

4.1 Experiments on Offline Data Set

In this section, we evaluate the performance of our proposed color-assisted L_1-subspace background modeling algorithm on a publicly available offline data set: the *Daniel* video sequence. The moving object in this video is a person walking back and forth. The first 10 frames of the video sequence is used to calculate the proposed pixel-wise RGB L_1-subspace followed by RGB background estimation. These frames are also used to estimate the gray-scale background. Afterwards, the moving object binary mask is generated by combining the color component detected mask and the gray-scale component detected mask. In Table 1, we compare the binary mask detected by the proposed method and the existing gray-scale L_1-PCA background modeling for two sample frames of the video sequence. It's observed that the proposed binary mask is much closer to the ground-truth

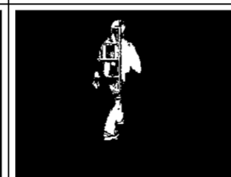
	Sample Frame	Ground Truth Mask	Proposed Mask	L1-PCA Mask
t=25				
Recall			0.973	0.630
Precision			0.973	0.988
Similarity			0.947	0.625
F1			0.973	0.770
t=37				
Recall			0.930	0.564
Precision			0.960	0.978
Similarity			0.896	0.557
F1			0.945	0.716

Table 1: Detected binary masks of the Daniel sequence.

binary mask than that detected by the gray-scale L_1-PCA method.[9] To provide quantitative analysis, we also calculate different accuracy metrics such as *Recall*, *Precision*, F_1, and *Similarity*. Let's denote the number of true positives, false negatives, and false positives as *tp*, *fn*, and *fp*, respectively. Then *Recall*, also known as the *detection rate*, is defined as

$$Recall \triangleq \frac{tp}{tp + fn}. \tag{8}$$

It calculates the percentage of the detected true foreground object pixels. *Recall* alone is not enough to compare different methods and is generally used in conjunction with *Precision*, also known as *positive prediction*, that gives the percentage of detected true positives as compared to the total number of foreground object pixels detected by the method,

$$Precision \triangleq \frac{tp}{tp + fp}. \tag{9}$$

Moreover, we consider the F_1 metric, also known as *Figure* of *Merit* or F-measure, that is the weighted harmonic mean of *Precision* and *Recall*,

$$F_1 \triangleq \frac{2 * Recall * Precision}{Recall + Precision}. \tag{10}$$

Such measure allows to obtain a single value that can be used to "rank" different methods. Finally, we consider the pixel-based *Similarity* measure defined as

$$Similarity \triangleq \frac{tp}{tp + fn + fp}. \tag{11}$$

Since the proposed method utilize both color information and luminance information of the video data, it offers much higher *Recall*, *Similarity*, and F_1 scores for both sample frames compared to the existing gray-scale L_1-PCA method.

4.2 Experiments on Real-time Camera Data

In this section, we carry out experiments on real-time camera data to evaluate the performance of the color-assisted moving object detection algorithm on the proposed edge-to-fog computing paradigm. Two Intel Core i7 laptops act as fog nodes that process the L_1-subspace computation on the video data. A Raspberry Pi 3B+ with an iHome USB 2.0 webcam acts as the edge node to capture video frames, compress them, and send them to the fog nodes. The Raspberry Pi represents the IoT device that produces large quantities of data. To route our Internet traffic, we use a Netgear N300 WiFi router. The testbed used in this research is shown in Fig. 2. Table 2 details the specifications of each of the hardware components.

System Specifications			
	CPU	RAM	OS
Fog Server 1	Intel i7 4720HQ @ 2.6 GHz	12 GB RAM	Windows 10, 64 bit
Fog Server 2	Intel i7 5500U @ 2.4 GHz	8 GB RAM	Windows 10, 64 bit
Raspberry Pi 3B+	ARM Cortex A53 @ 1.4 GHz	1 GB RAM	Raspbian Stretch
Camera	iHome IH-W312NP USB 2.0 Mylife Notebook Webcam		
Router	NETGEAR N300		

Table 2: Specifications of equipment used in the proposed edge-to-fog computing platform.

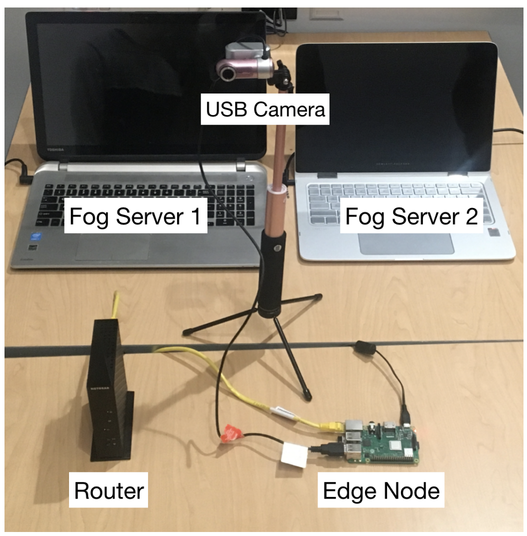


Figure 2: The testbed of the proposed edge-to-fog computing platform.

Fig. 3 shows four sample frames taken by the camera at time slot $t = 7, 10, 13$, and 19, and the corresponding binary masks for the moving objects detected by the three algorithms in comparison. By comparing the binary masks of the RGB only (Fig. 3 (c)) and gray-scale only (Fig. 3 (d)) background estimations to the proposed algorithm results (Fig. 3 (b)), it is evident that the proposed method is more accurate in identifying the objects in the original frames (Fig. 3 (a)). As such, the proposed method that combines the RGB L_1-subspace background estimation with the gray-scale background estimation, we are able to achieve improved moving-object detection through color-assistance that takes advantage of the pixel-wise L_1-subspace learning.

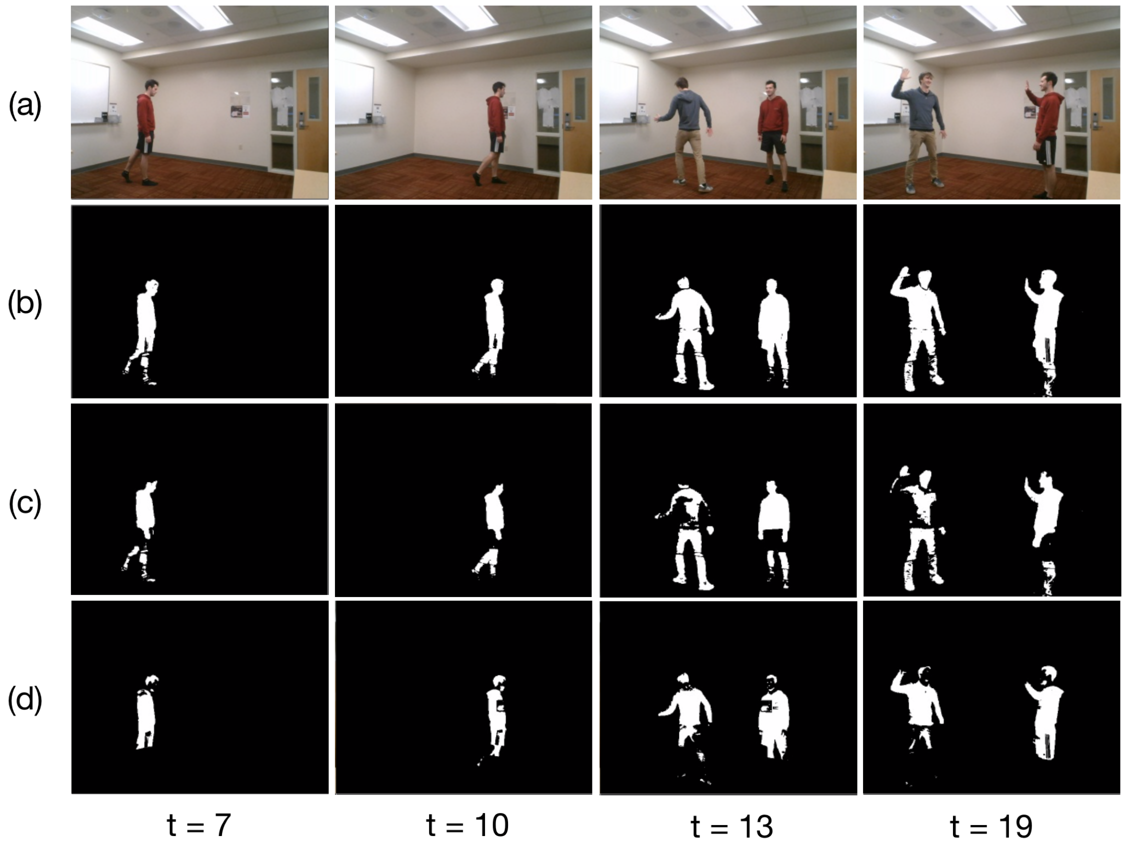


Figure 3: Sample frames of real-time camera data: (a) original frames; detected binary masks of (b) the proposed color-assisted algorithm; (c) RGB only background estimation; and (d) gray-scale only background estimation.

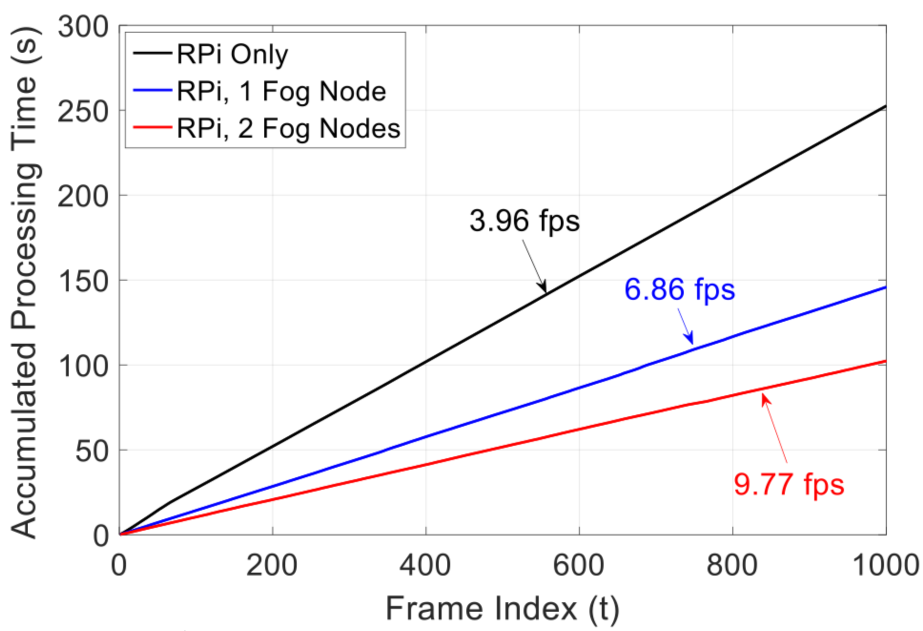


Figure 4: Accumulated processing time of varying computing platforms.

In addition, we measure the performance of various hardware configurations in executing the proposed algorithm to demonstrate the effectiveness of our proposed edge-to-fog architecture in providing low-latency moving object detection. Fig. 4 provides a measure of the accumulated processing time of three different computing architectures for a fixed amount of video frames. The use of a pure edge architecture consisting of a single Raspberry Pi proves to take the longest accumulated time to process 1000 frames. In turn, the use of a single Raspberry Pi in conjunction with an i7 for server in an edge-to-fog network shows dramatic improvement in accumulated processing time of 1000 images. By adding a second i7 fog server to this edge-to-fog network and splitting each captured video frame into two sub-frames for processing, the accumulated processing time of 1000 frames further reduces, as demonstrated by the line with the most gradual slope in Fig. 4.

Based on the experimental results, the use of an edge-to-fog network is clearly most efficient in processing and displaying the binary masks of the moving objects. By adding fog nodes to the architecture and splitting each video frame into an equivalent number of sub-frames for processing, the speed of computation is improved. As such, the edge-to-fog network is most effective for producing results in a timely manner on a local network. Based on this data, the use of the edge-to-fog architecture allows for the accuracy of the color-assisted L_1-subspace algorithm to be combined with the speed of a local network, both of which are necessary for IoT applications of low-latency moving object detection.

5. CONCLUSION

In this work, we have demonstrated the improvement of moving object detection through the use of the RGB color-space in the L_1-subspace algorithm. In our proposed algorithm, the combination of the RGB L_1-subspace background estimation with the gray-scale background estimation generates improved binary masks of the moving objects. In addition, an edge-to-fog system was designed to accommodate the speed and accuracy demands of IoT devices. Through the implementation of color-assisted moving object detection on an edge-to-fog paradigm, our research provides a novel solution for low-latency, high-accuracy video processing systems required by ever-advancing IoT technology.

In the future, the results of applying different color-spaces to the L_1-subspace algorithm can be combined to further improve the accuracy of video background modeling. In addition, the edge-to-fog computing paradigm can be adapted to meet the low-latency demands of future IoT devices by adding fog nodes, introducing process threading, increasing the computational power of the various nodes, and other architectural modifications. This project provides a foundation for future advancements in the speed and accuracy of IoT video data processing by demonstrating the improvement of moving object detection through the use of color-spaces and an edge-to-fog computing architecture.

REFERENCES

[1] Liu, Y., Yao, H., Gao, W., Chen, X., and Zhao, D., "Nonparametric background generation," *J. Visual Commun. and Image Representation* **18**(3), 253–263 (2007).

[2] Haque, M., Murshed, M., and Paul, M., "Improved gaussian mixtures for robust object detection by adaptive multi-background generation," *Proc. IEEE Int. Conf. Pattern Recognition* , 1–4 (2008).

[3] Zheng, Y. and Fan, L., "Moving object detection based on running average background and temporal difference," *Proc. IEEE Int. Conf. Intell. Syst. and Knowl. Eng. (ISKE)* , 270–272 (2010).

[4] Bouwmans, T., Porikli, F., Höferlin, B., and Vacavant, A., [*Background modeling and foreground detection for video surveillance*], CRC press (2014).

[5] Seo, J.-W. and Kim, S. D., "Recursive on-line $(2d)^2$pca and its application to long-term background subtraction," *IEEE Trans. Multimedia* **16**(8), 2333–2344 (2014).

[6] Markopoulos, P. P., Karystinos, G. N., and Pados, D. A., "Optimal algorithms for l_1-subspace signal processing," *IEEE Trans. Signal Process.* **62**(19), 5046–5058 (2014).

[7] Kundu, S., Markopoulos, P. P., and Pados, D. A., "Fast computation of the l_1-principal component of real-valued data," *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process. (ICASSP)* , 8028–8032 (2014).

[8] Markopoulos, P. P., Kundu, S., Chamadia, S., and Pados, D. A., "Efficient l1-norm principal-component analysis via bit flipping," *IEEE Trans. Signal Process.* **65**(16), 4252–4264 (2017).

[9] Liu, Y. and Pados, D. A., "Compressed-sensed-domain l_1-pca video surveillance," *IEEE Trans. Multimedia* **18**(3), 351–363 (2016).

[10] Pierantozzi, M., Liu, Y., Pados, D. A., and Colonnese, S., "Video background tracking and foreground extraction via l1-subspace updates," *Proc. SPIE Commercial + Scientific Sensing and Imaging, Compressive Sensing V: From Diverse Modalities to Big Data Analytics* **9857**, 985708 (2016).

[11] Liu, Y., Pados, D. A., Batalama, S. N., and Medley, M. J., "Iterative re-weighted l_1-norm principal-component analysis," *Proc. IEEE Asilomar Conf. Signals, Syst., and Comput.* , 425–429 (2017).

[12] Liu, Y. and Pados, D. A., "Conformity evaluation of data samples by l_1-norm principal-component analysis," *Proc. SPIE Commercial + Scientific Sensing and Imaging, Compressive Sensing VII: From Diverse Modalities to Big Data Analytics* **10658**, 1065809 (2018).

[13] Mohan, A., Gauen, K., Lu, Y.-H., Li, W. W., and Chen, X., "Internet of video things in 2030: A world with many cameras," *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)* , 1–4 (2017).

[14] Satyanarayanan, M., "The emergence of edge computing," *Computer* **50**, 30–39 (Jan 2017).

[15] Khan, N. N., "Fog computing: A better solution for iot," *Int. J. Eng. and Tech. Research* **3**(2), 298–300 (2015).

[16] Soyata, T., Muraleedharan, R., Funai, C., Kwon, M., and Heinzelman, W., "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," *Proc. IEEE Symp. Comput. Commun. (ISCC)* , 59–66 (2012).

[17] Hu, P., Ning, H., Qiu, T., Zhang, Y., and Luo, X., "Fog computing based face identification and resolution scheme in internet of things," *IEEE Trans. Ind. Inform.* **13**(4), 1910–1920 (2017).

[18] Muslim, N. and Islam, S., "Face recognition in the edge cloud," *Proc. ACM Int. Conf. Imaging, Signal Process. and Commun.* , 5–9 (2017).

[19] Kwak, N., "Principal component analysis based on l_1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(9), 1672–1680 (2008).

[20] Nie, F., Huang, H., Ding, C., Luo, D., and Wang, H., "Robust principal component analysis with non-greedy l_1-norm maximization," *Proc. Int. Joint Conf. on Artificial Intell.* , 1433–1438 (2011).

[21] Markopoulos, P. P., Karystinos, G. N., and Pados, D. A., "Some options for l1-subspace signal processing," *Proc. of the Tenth Int. Symp. Wireless Commun. Syst. (ISWCS)* , 1–5 (2013).

[22] Maritato, F., Liu, Y., Colonnese, S., and Pados, D. A., "Face recognition with l_1-norm subspaces," *Proc. SPIE Commercial + Scientific Sensing and Imaging, Compressive Sensing V: From Diverse Modalities to Big Data Analytics* **9857**, 98570L (2016).

[23] Maritato, F., Liu, Y., Colonnese, S., and Pados, D. A., "Cloud-assisted individual l_1-pca face recognition using wavelet-domain compressed images," *Proc. the 6th European Workshop on Visual Inform. Process. (EUVIP)* , 1–6 (2016).