An Adaptive Video Acquisition Scheme for Object Tracking

Srutarshi Banerjee[1*], Juan G. Serra[2], Henry H. Chopp[1], Oliver Cossairt[3], A. K. Katsaggelos[1],

[1] Dept. of Electrical and Computer Engineering, Northwestern University, Illinois, USA.
[2] Dpto. de Ciencias de la Computación e I.A., Universidad de Granada, Granada, Spain
[3] Dept. of Computer Science, Northwestern University, Illinois, USA.
[*]e-mail:srutarshibanerjee2022@u.northwestern.edu

***Abstract*—In this paper, we propose an adaptive host-chip system for video acquisition constrained under a given bit rate to optimize object tracking performance. The chip is an imaging instrument with limited computational power consisting of a very high-resolution focal plane array (FPA) that transmits quadtree (QT)-segmented video frames to the host. The host has unlimited computational power for video analysis. We find the optimal QT decomposition to minimize a weighted rate distortion equation using the Viterbi algorithm. The weights are user-defined based on the class of objects to track. Faster R-CNN and a Kalman filter are used to detect and track the objects of interest respectively. We evaluate our architecture's performance based on the Multiple Object Tracking Accuracy (MOTA).**

***Index Terms*—host-chip architecture, Viterbi algorithm, optimal bit allocation, rate distortion, object tracking**

I. Introduction

In this work, we focus on the problem of optimal information extraction in wide-area surveillance imaging applications using high resolution sensors. We assume an imaging instrument of limited computational power consisting of a Focal Plane Array (FPA). The FPA provides fixed or moving viewpoint imagery over the desired field of view. The goal is to autonomously acquire the scene so that the spatio-temporal information is preserved for detection and tracking of objects of interest. This analysis is done on a separate host device with unlimited computational resources. The first challenge is that the bandwidth of the Readout Integrated Circuit (ROIC) limits the maximum number of bits/s that may be delivered from the sensor to the host. Hence, the problem bears close similarity to resource allocation problems faced in image/video compression and transmission literature [1], [2], [3]. The second challenge is to perform object tracking on the distorted video sequence due to the limited bandwidth. This makes tracking more challenging than when a frame is analyzed in its original state.

In today's commercial FPA technology, a variety of controls over the spatio-temporal sampling properties of the sensor is available. Pixel-binning and sub-sampling modes allow for a dynamic tradeoff between spatial and temporal resolutions. High frame rates (e.g. $>$ 1 kfps) may be achieved at low resolution (e.g. $<$ VGA), while maximum frame rates that can be achieved for high resolution FPAs (e.g. $>$ 10 MPix) are typically low ($<$ 60 Hz). These pixel readout modes provide a way to optimize sampling performance given constraints on the maximum pixel bandwidth of the ROIC electronics [4], [5], [6], [7], [8], [9].

Adaptivity during acquisition has been introduced in different ways. For instance, local features are extracted in the measurement domain, such as standard deviation [10], [11], edge counting [12] or estimation of the reconstruction error [13], so as to guide the adaptive acquisition. Similarly, an adaptive scheme is proposed in [14] by estimation of the compressibility based on the local redundancy which is measured statistically utilizing previously sensed measurements. These measurement domain based recovery algorithms can benefit from on-the-fly adaptive sampling as they do not require a feedback channel from the receiver side but are often not so accurate in the reconstruction.

Object detection and tracking is a difficult problem that has been addressed by many researchers, more recently by using neural networks to perform the task. Howard et al. [15] proposed a lightweight object detector designed for resource-constrained mobile platforms. R-FCN [16] using region based, fully convolutional neural network based on ResNets [17] as backbone has also been proposed. Tubelets [18] were designed using a convolutional neural network (CNN) to perform both object detection and tracking given a video sequence in full. Many online trackers build appearance models of either the individual objects themselves [19] or through a global model [20]. These networks were essentially trained with actual data and not on distorted data.

To the best of the authors' knowledge, no previous work has addressed the adaptive sampling of the spatio-temporal volume at once along with the reconstruction algorithms designed for object tracking. Our approach, therefore, is the first to extend existing results in adaptive sensing and consider a comprehensive approach to the flexible asynchronous space-time image acquisition problem. Our assessment goal for this algorithm is not the traditional reconstructed image quality (e.g., PSNR, SSIM), but rather the tracking performance of objects of interest.

This paper is organized as follows: in Section II, we formulate the problem. Section III describes our architecture

This work was supported in part by a DARPA Contract No. HR0011-17-2-0044

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

for tracking. Section IV describes our experimental results and Section V concludes the paper.

II. Problem Formulation

The proposed work focuses on developing a methodology for adaptive guidance of a sensor through real-time tuning of sensor control parameters to collect data with the highest content of useful information for object tracking. It is based on a computational imaging approach using a prediction-correction paradigm. The goal is to use the host for predicting an optimal sampling pattern on the chip and correcting that predicted sampling strategy. The host computer helps guide the sampling strategy for the chip, consisting of the FPA and ROIC, so that the host can optimally perform object tracking with its unlimited computational power.

The proposed architecture will allow dynamic, reconfigurable, and content-adaptive sensing of spatio-temporal information with optimal bandwidth utilization. We pose the optimization problem as a resource allocation problem: given the constraint on the allowable data bandwidth connecting the sensor and the host computer, we estimate the best possible tessellation per frame. This reduces the number of bits required to represent a frame. The communication channel between the host and chip can best utilize the bandwidth by only sending important information.

The adaptive segmentation of the video frame is data-driven based on a quadtree (QT) structure. The chip designs a QT structure that subdivides the current frame into superpixels before transmitting it to the host. Superpixels with high distortion may only be subdivided based on the available bandwidth. If there is available bandwidth, the QT may further divide to capture finer details in a video frame. QTs for newly acquired frames on the chip contain information about the superpixels the host should update or ignore (skip) in its frame of the previous time step. The superpixel intensities for the update regions are sent to the host. Skipped superpixels assume the previous value. The QT is designed based on: (i) the distortion between the current frame and the previous reconstructed frame, and (ii) the predicted locations of the objects of interest for the current frame. A fast and effective recursive encoding of the QT structure is developed in [21].

III. Host-Chip Architecture

The host-chip architecture shown in Fig. 1 and Fig. 2, respectively, works as a prediction–correction model. This architecture was designed keeping in mind the limited computational resources on the chip and high computational power on the host. The host predicts the location of the Regions of Interest (ROIs) for frame f_{t+1}. Based on the next frame QT sent from the chip, the host then corrects those predictions. To best track the ROIs under the bandwidth constraint between the host and the chip, B, the reconstructed image on the host has higher resolution for the ROIs and poorer resolution for the rest of the frame.

For a QT acquisition of frame f_t on the chip, S_t, we have a skip-acquire mode for the leaves of the QT, Q_t, and pixel values V_t for the leaves of acquire modes. These are sent from the chip to the host to create the reconstructed frame $\hat{f}_t$. Values in skip leaves are copied from the previously reconstructed frame $\hat{f}_{t-1}$ available on the host. Using an object detector, the host then determines the ROIs of the reconstructed image $\hat{f}_t$. The ROIs are fed into an object tracker which predicts the next ROIs for frame f_{t+1}, denoted as $\widetilde{bb}_{t+1}$. The predicted ROIs for frame f_{t+1} are sent to the chip.

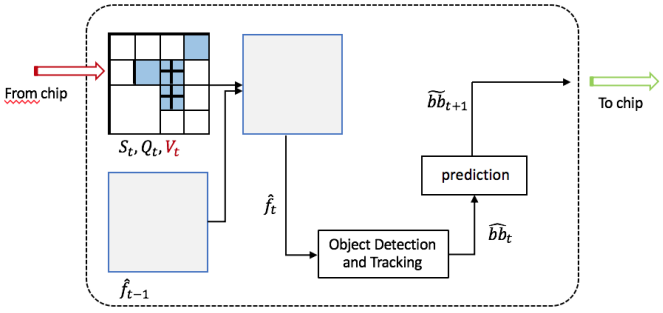

Figure 1: Computation on Host

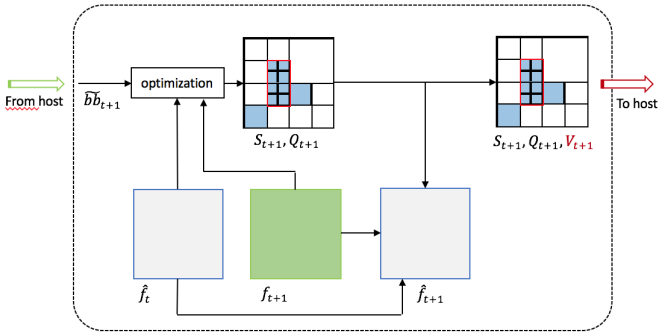

Figure 2: Computation on Chip

The chip receives $\widetilde{bb}_{t+1}$ and possesses $\hat{f}_t$. The full resolution frame f_{t+1} is acquired at time $t+1$ from the FPA. The QT structure for $\hat{f}_{t+1}$ is found using a Viterbi optimization, described in Subsection A. The Viterbi algorithm provides the optimal S_{t+1} and Q_{t+1} subject to the bandwidth constraint B. This information, along with corresponding pixel values V_{t+1}, are sent to the host in order to reconstruct and analyze the frame $\hat{f}_{t+1}$.

A. Viterbi Optimization

The goal of the optimization is to have a trade-off between the frame rate and frame distortion by minimizing the frame distortion D over the leaves of the QT $\mathbf{x}$ subject to a given maximum frame rate R_{max}. In previous works [21], [22], [23], the Viterbi Optimization has been used for compression with the actual frames. However, in this work, while computing frame distortion and rate, the reconstructed frame $\hat{f}_t$ is taken as an input along with the actual frame f_{t+1} acquired by the FPA and the ROIs predicted in frame $t+1$ by the Kalman

Tracker. The Viterbi algorithm is used to estimate the optimal $\hat{f}_{t+1}$ for object tracking purposes.

The optimization is formulated in the following way as shown in Eqn. 1:

$$\begin{aligned} &\arg\min_{\mathbf{x}} \ D(\mathbf{x}), \\ &\text{s. t. } R(\mathbf{x}^*) \leq R_{max}. \end{aligned} \tag{1}$$

The distortion for each node of the QT is based on the acquisition mode Q_t of that node. If the node $\hat{x}_t$ at particular reconstructed frame at time t is skip, the distortion with respect to the new node at time $t+1$, x_{t+1} is given in Eqn. 2:

$$D_s = |x_{t+1} - \hat{x}_t|. \tag{2}$$

On the other hand, if the tree node is acquire, the distortion is proportional to the standard deviation σ. This is shown in Eqn. 3, where N is the maximum depth of the QT and n is the level of the QT where distortion is computed. The root is defined to be on level 0, and the most subdivided level as N:

$$D_a = \sigma \times 4^{N-n}. \tag{3}$$

The total distortion is therefore defined as

$$D = D_s + D_a. \tag{4}$$

The constrained discrete optimization of Eqn. 1 is solved using Lagrangian relaxation, leading to solutions in the convex hull of the rate-distortion curve [22]. The Lagrangian cost function is of the form

$$J_\lambda(\mathbf{x}) = D(\mathbf{x}) + \lambda R(\mathbf{x}) \tag{5}$$

where $\lambda \geq 0$ is a Lagrangian multiplier. It has been shown that if there is a λ^* such that

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} J_{\lambda^*}(\mathbf{x}) \tag{6}$$

which leads to $R(\mathbf{x}^*) = R_{max}$, then $\mathbf{x}^*$ is the optimal solution to Eq. 1. This is solved using the Viterbi algorithm, shown in detail in [21]. However, we want to prioritize the regions based on the bounding boxes, which are the ROIs. The following cost function is a modification of Eqn. 5:

$$J_\lambda(\mathbf{x}) = \sum_{i \in \Omega} w_i D_i(\mathbf{x}_i) + \lambda R(\mathbf{x}) \tag{7}$$

where, Ω: set of differently weighted regions. D_i: distortion of region i. w_i: weights of region i.

The bit rate in each frame can be fixed at a desired bit rate within a certain tolerance. This is done by adjusting λ in the Lagrangian multiplier method. This optimal λ^* is arrived at using a convex search based on a Bezier curve [22], which accelerates convergence.

B. Object Detection and Tracking

The Viterbi Algorithm is fed with the predicted bounding boxes for the next frame. This is done with the combination of the object detector and tracker as shown in Fig. 1.

A Convolutional Neural Network (CNN)-based object detector, Faster R-CNN [24], has been used to detect objects of interest. The Faster R-CNN comprises of two modules: the first module consists of a deep fully convolutional network which proposes the regions of probable objects, and the second module classifies the objects in those region proposals. The object detector is located on the host with access to only the distorted reconstructed frames. In order to enhance the performance of the Faster R-CNN for degraded data as well, the object detector has been trained on both non-distorted and distorted data. Three classes of objects (airplanes, cars and watercraft) from the ILSVRC VID dataset [25] were used to train the Faster R-CNN network. The original video frames from this data subset have a distortion applied corresponding to $\lambda = 10$ and $\lambda = 150$ by passing the frames through the proposed architecture. The ground truth bounding boxes are used in place of the object detector to generate realistic distortions that the model will encounter.

The object detector generates bounding boxes with class labels. The bounding boxes are the inputs to a tracker. A Kalman Filter-based multiple object tracker, Simple Online and Realtime Tracking (SORT) [26] is used. It first predicts the future bounding box locations using a linear motion model. Then, it associates the identities using linear assignment between the new detections from the object detector and the most recently predicted bounding boxes. The state of the Kalman Filter, $\mathbf{X}_s$ for each detection is modeled using a linear motion model as

$$\mathbf{X}_s = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \tag{8}$$

where u and v represent the horizontal and vertical location of the center of the target, s and r represent the scale (area) and the aspect ratio (width/height) of the target's bounding box respectively. Three of these time derivatives are a part of the state as well: $\dot{u}$, $\dot{v}$, and $\dot{s}$; it assumes that the aspect ratio is constant.

When a detection is associated with a target, the detected bounding box is used to update the target state where the velocity components are solved optimally via the Kalman filter framework [27]. The predicted bounding boxes are extracted from the predicted state of the Kalman filter. These bounding boxes are the ROIs for the acquisition of the next frame $t+1$ which are also inputs to the Viterbi algorithm.

C. Performance Accuracy Metric

In order to evaluate the multi-target performance, we utilize the Multiple Object Tracking Accuracy (MOTA) evaluation metrics defined in [28]:

$$MOTA = 1 - \sum_t \frac{m_t + fp_t + mme_t}{g_t}, \tag{9}$$

where m_t: number of missed detections at time t. fp_t: number of false positives at time t. mme_t: number of mismatch (track switching) errors at time t. g_t: number of ground truth objects at time t.

Evaluation measures with a higher MOTA score correspond to a better performance. The experiments are conducted for different values of fixed λ. This gives different average bit rates over a video sequence, which are a fraction of the maximum rate. λ provides the operating point in the rate-distortion curve. For different values of λ, the distortion in each frame is kept constant while the bit rate fluctuates for each frame.

The priority weights w_i of Eqn. 7 are assigned by us in the Viterbi algorithm by object class in order to allocate more bits in ROIs with high priority and fewer bits to the regions of lower priority. Fine sampling of the image occurs in areas where we allocate higher weights, whereas in the remaining areas pixels are averaged with coarse resolution.

IV. Experimental Results

Our preliminary results are generated by simulating the proposed model on three sequences of the ILSVRC VID dataset: (i) a video of airplanes, ILSVRC2015_val_00007010.mp4; (ii) a video of a watercraft, ILSVRC2015_val_00020006.mp4; and (iii) a video of cars, ILSVRC2015_val_00144000.mp4. These videos were resized to 512×512 to accommodate the QT structure. The results were generated using $w_{Obj} = 1000$ and $w_{Backgnd} = 100$ in reference to Eqn. 7.

We compare the results of two neural network (NN) models: (i) the Faster R-CNN trained exclusively with pristine (non-distorted) data of the three classes (Pristine NN model), and (ii) the Faster R-CNN trained with pristine data and distorted data for $\lambda = 10$ and $\lambda = 150$ (Mixed NN model). The NN models were trained using ADAM as the optimizer with a learning rate of $1e-5$. Dropout rate of 0.5 was used while training both of the models.

Airplane Sequence			
λ	Bit Rate Compression	MOTA (Mixed)	MOTA (Pristine)
0	3.5523	0.7223	0.6429
10	18.6599	0.7404	0.6590
150	59.5121	0.6338	0.6187

Table I: ILSVRC2015_val_00007010.mp4 results

Watercraft Sequence			
λ	Bit Rate Compression	MOTA (Mixed)	MOTA (Pristine)
0	1.2414	0.8117	0.6883
10	3.3226	0.7922	0.7078
150	37.5215	0.7532	0.5260

Table II: ILSVRC2015_val_00020006.mp4 results

Car Sequence			
λ	Bit Rate Compression	MOTA (Mixed)	MOTA (Pristine)
0	2.6691	0.6516	0.5342
10	10.1177	0.6091	0.5413
150	49.0622	0.4126	0.4661

Table III: ILSVRC2015_val_00144000.mp4 results

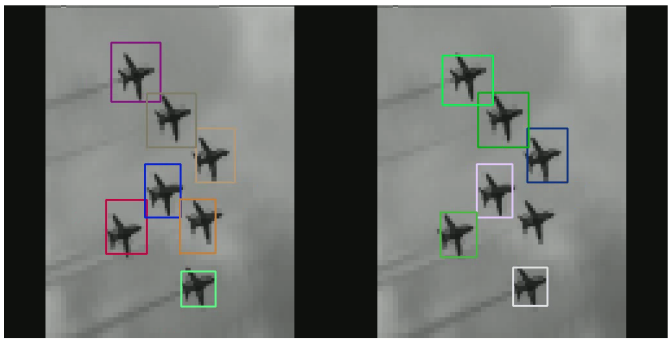

Figure 3: $\hat{f}_{52}$ of the Airplance Sequence for $\lambda = 150$ (left) Mixed NN (right) Pristine NN

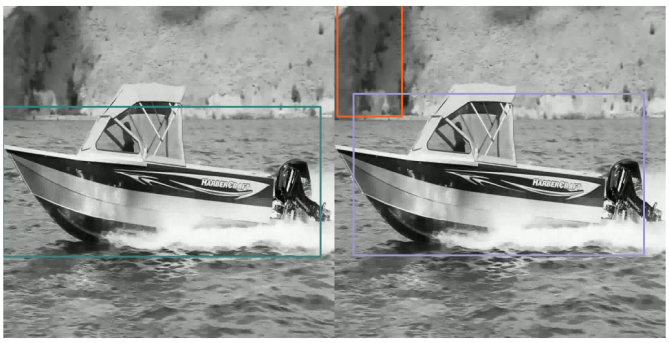

Figure 4: $\hat{f}_{54}$ of the Watercraft Sequence for $\lambda = 10$ (left) Mixed NN (right) Pristine NN

The results of the Pristine NN and Mixed NN models are tabulated in Tables I, II, and III. The bit rate compression for the test sequences at different λ values are also shown. As λ increases from 0 to 10 to 150, the compression factor also increases. For $\lambda = 0$, the distortion is 0, however, depending on the scene, not all pixel values need to be sent to the host. Thus, we are even able to achieve a lossless compression for non-distorted videos.

For the test sequences, the Mixed NN Model well outperformed the Pristine NN Model except in the car Sequence when $\lambda = 150$. We see as an example in Figure 3 that missed

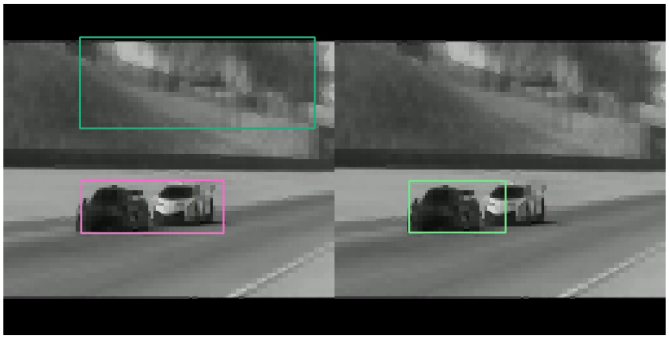

Figure 5: $\hat{f}_{48}$ of the Car Sequence for $\lambda = 150$ (left) Mixed NN (right) Pristine NN

detections of the Pristine NN Model are more prevalent for the Airplane Sequence than its Mixed NN Model counterpart. In Figure 4, the Pristine NN Model produces a false positive in the top left corner. Lastly, we see two errors in the Mixed NN Model: one false positive, and one missed detection. This occurs in multiple frames, which accounts for the relatively low MOTA score. Overall, however, we have shown that training with degraded data generally has improvements to the tracking results.

V. Conclusion

This paper proposes a new method of video acquisition for object tracking. ROIs influence how the architecture processes new frames and updates QT structures. A predictive Viterbi-based optimization was used to generate the FPA's acquisition modes and the optimal QT structure that minimizes the weighted rate-distortion equation. A Faster R-CNN object detector was trained with pristine and distorted data to improve the tracking performance. A Kalman filter-based tracker was used to track detected objects. Preliminary experiments provide strong support of the effectiveness of this method.

References

[1] G. M. Schuster, and A. K. Katsaggelos, Rate-Distortion Based Video Compression: Optimal Video Frame Compression and Object Boundary Encoding, Springer, 1996.

[2] F. Zhai and A. K. Katsaggelos, "Joint Source-Channel Video Transmission," Synthesis Lectures on Image, Video, and Multimedia Processing, Morgan Claypool, 2006.

[3] A. K. Katsaggelos, Y. Eisenberg, F. Zhai, R. Berry, and T. N. Pappas, "Advances in Efficient Resource Allocation for Packet-Based Real-Time Video Transmission," IEEE Proceedings, vol. 93, pp. 135-147, Jan. 2005.

[4] R. Koller et al., "High Spatio-Temporal Resolution Video with Compressed Sensing," Opt. Express vol. 23, Iss. 12, pp. 15992-16007, 2015.

[5] L. Spinoulas, O. Cossairt and A. K. Katsaggelos, "Sampling optimization for on-chip compressive video," 2015 IEEE Int. Conf. on Image Processing (ICIP), Quebec City, QC, 2015, pp. 3329-3333.

[6] L. Spinoulas, K. He, O. Cossairt and A. Katsaggelos, "Video compressive sensing with on-chip programmable subsampling," 2015 IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, 2015, pp. 49-57.

[7] M. Gupta, A. Agrawal, A. Veeraraghavan and S. G. Narasimhan, "Flexible voxels for motion-aware videography," European Conf. on Computer Vision, Springer Berlin Heidelberg, 2010, pp. 100-114.

[8] D. Reddy, A. Veeraraghavan and R. Chellappa, "P2C2: Programmable pixel compressive camera for high speed imaging," CVPR 2011, Colorado Springs, CO, USA, 2011, pp. 329-336.

[9] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga and S. K. Nayar, "Video from a single coded exposure photograph using a learned over-complete dictionary," 2011 Int. Conf. on Computer Vision, Barcelona, 2011, pp. 287-294.

[10] I. Noor and E. L. Jacobs, "Adaptive compressive sensing algorithm for video acquisition using single pixel camera," SPIE J. Electronic Imaging, vol. 22, no. 2, pp. 021013–021013, Jul. 2013.

[11] W. Guicquero, A. Verdant, A. Dupret, and P. Vandergheynst, "Nonuniform sampling with adaptive expectancy based on local variance," Proc. Int. Conf. Sampling Theory and Applications (SampTA), Washington, DC, May 2015, pp. 254–258.

[12] W. Guicquero, A. Dupret and P. Vandergheynst, "An adaptive compressive sensing with side information," 2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 2013, pp. 138-142.

[13] D.M. Malioutov, S.R. Sanghavi, and A.S.Willsky, "Sequential compressed sensing," IEEE J. Sel. Topics in Signal Processing, vol. 4, no. 2, pp. 435–444, Apr 2010.

[14] J. Chen, X. Zhang, and H. Meng, "Self-adaptive sampling rate assignment and image reconstruction via combination of structured sparsity and non-local total variation priors," Digital Signal Processing, vol. 29, pp. 54–66, June 2014.

[15] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications ," arXiv: 1704.04861[cs], Apr 2017.

[16] J. Dai, Y. Li, K. He and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," arXiv: 1605.06409[cs], Jun 2017.

[17] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[18] K. Kang et al., "T-CNN: Tubelets With Convolutional Neural Networks for Object Detection From Videos," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 10, pp. 2896 - 2907, Aug. 2017.

[19] S. Bae and K. Yoon, "Robust Online Multi-object Tracking Based on Tracklet Confidence and Online Discriminative Appearance Learning," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1218-1225.

[20] A. Bewley, V. Guizilini, F. Ramos and B. Upcroft, "Online self-supervised multi-instance segmentation of dynamic objects," 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014, pp. 1296-1303.

[21] G. M. Schuster and A. K. Katsaggelos, "A Video Compression Scheme with Optimal Bit Allocation among Segmentation, Motion, and Residual Error," in IEEE Trans. on Image Processing, vol. 6, no. 11, pp. 1487-1502, Nov. 1997.

[22] G. M. Schuster and A. K. Katsaggelos, "An optimal quadtree-based motion estimation and motion-compensated interpolation scheme for video compression," in IEEE Transactions on Image Processing, vol. 7, no. 11, pp. 1505-1523, Nov. 1998.

[23] E. Soyak, S. A. Tsaftaris and A. K. Katsaggelos, "Low-Complexity Tracking-Aware H.264 Video Compression for Transportation Surveillance," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 10, pp. 1378-1389, Oct. 2011.

[24] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.

[25] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge ," in Int. Jour. of Computer Vision, vol. 115, no. 3, pp. 211 - 252, 2015.

[26] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple Online and Realtime Tracking," arXiv:1602.00763 [cs], Jul. 2017.

[27] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Jour. of Basic Eng., vol. 82, no. D, pp. 35 - 45, 1960.

[28] K. Bernardin, A. Elbs, and R. Stiefelhagen, "Multiple Object Tracking Performance Metrics and Evaluation in a Smart Room Environment," in Sixth IEEE International Workshop on Visual Surveillance, May 2006, Graz, Austria.