ABANDONED OBJECT DETECTION USING PIXEL-BASED FINITE STATE MACHINE AND SINGLE SHOT MULTIBOX DETECTOR

*Devadeep Shyam, and Alex Kot**

Rapid-Rich Object Search (ROSE) Lab
Nanyang Technological University, Singapore
dshyam@ntu.edu.sg, and eackot@ntu.edu.sg

Chinmayee Athalye

Electronics And Telecommunication Engg.
College of Engineering Pune, India
chinmayeeathalye@outlook.com

ABSTRACT

This paper proposes a robust, scalable framework for automatic detection of abandoned, stationary objects in real time surveillance videos that can pose a security threat. We use the sViBe background modeling method to generate a long-term and a short-term background model to extract foreground objects. Subsequently, a pixel-based FSM detects stationary candidate objects based on the temporal transition of code patterns. In order to classify the stationary candidate objects, we use deep learning method (SSD: Single Shot MultiBox Detector) to detect person and some suspected type of objects which include backpack, handbag. In order to suppress any false alarm, we remove other stationary candidate objects other than the suspected stationary objects. After stationary object detection, we also check if there is no person near by the suspected detected objects for a particular time. We tested the system on four standard public datasets. The results show that our method outperforms the performance of existing results while also being robust to temporary occlusions and illumination changes.

***Index Terms*—** SSD: Single Shot MultiBox Detector, ViBe, sViBe, GMM, SILTP

1. INTRODUCTION

Detection of abandoned objects is a crucial task to ensure public security. Due to the increasing number of surveillance devices and the ensuing data flood, it has become impossible to manually process all the feeds from a surveillance camera. Automatic detection of abandoned objects is one step towards providing better measures for public safety.

1.1. Related Work

The approaches used for detection of abandoned objects include three basic steps background modeling, stationary object detection, and object classification and/or tracking. One of the most successful background modeling methods till present is Gaussian Mixtures Model (GMM). It was first proposed by Stauffer and Grimson [1] in 1999. Lin [2], Heras Evangelio *et al.* [3] use GMM and its variations for abandoned object detection. GMM models every pixel as a mixture of K Gaussian functions. Each pixel is classified as background or foreground based on the Gaussian distribution that represents it most effectively. A high number of parameters need to be specified in this model and it is challenging to compute the values of these parameters for real-life, noisy environments and changing scene conditions. Barnich and Droogenbroeck [4] proposed the Visual Background Extractor (ViBe) as a universal background subtraction algorithm for video sequences. The superior performance of ViBe has led to many enhancements being proposed and adapted for background subtraction [5], [6], [7], [8]. ViBe uses only colour values of pixels to build the background model. As the colour values are affected by noise and illumination changes, performance of ViBe is easily affected by these factors.

Stationary foreground detection can be categorized into two main types of methods - double background models, and tracking foreground regions. Double background models make use of varying learning rates of the background modeling method to generate a long-term and a short-term background model. The difference between these models is used to obtain the static foreground [9], [10]. The problem with using direct difference between the two models is that it leads to a high false alarm rate. The way around this is to use a finite state machine based on the pixel values of the long-term and short-term models [2], [3], [11]. The second category involves tracking foreground over a period to detect stationary foreground objects. Xu *et al.* [12] and Martinez-del-Rincon *et al.* [13] use the Kalman Filter for tracking the foreground objects. Tripathi *et al.* [14] use blob association with bipartite matching for tracking. These methods fail in more complex scenes and where the stationary foreground is not clear.

Once the stationary foreground objects have been detected, they need to be classified as stationary persons and other foreground objects that form stationary foreground.

*This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Singapore, and the Infocomm Media Development Authority, Singapore.

978-1-5386-1737-3/18/$31.00 ©2018 IEEE

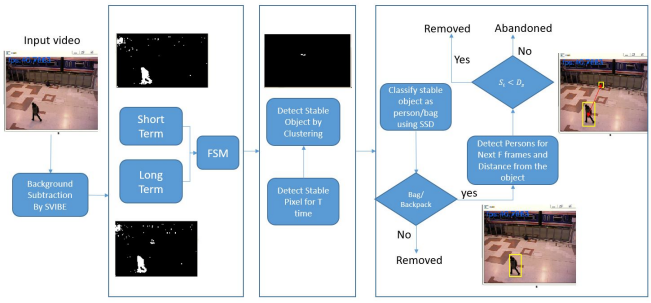


Fig. 1: The proposed methodology of our framework

Martnez-del-Rincn *et al.* [13] use homographic transform and height estimation of humans to distinguish between stationary persons and objects. Lin *et al.* [2] use the deformable part-based model to detect humans in the foreground scene. Tripathi *et al.* [14] use template matching to generate a score to decide whether the static object is human or non-human. General purpose object detection should be fast, accurate, and able to recognize a wide variety of objects. Since the introduction of neural networks, detection frameworks have become increasingly fast and accurate such as Fast RCNN, Faster RCNN, YOLO, SSD [15]. While Fast RCNN, Faster RCNN are accurate, but these approaches have been too computationally intensive for embedded systems and, even with high-end hardware, too slow for real-time applications. As mentioned in [15], SSD with a 300×300 input size significantly outperforms its 448×448 YOLO counterpart in accuracy and speed. SSD is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.

1.2. Our Approach

In this paper, we use sViBe [12] method for background estimation. We keep two update speeds of the sViBe background model to generate a long-term and a short-term background model. The pixel values of these two models are given as inputs to a pixel-based Finite State Machine (PFSM) which uses the temporal transition information from the input to detect stationary candidate foreground objects. This FSM has been designed to robustly detect temporarily occluded stationary candidate foreground objects. We classify the detected stationary candidate objects to further verify for stationary suspected objects and employ an abandonment check using predefined spatio-temporal rules.

2. SYSTEM DESCRIPTION

2.1. Background Modeling

We use sViBe as our method for background modeling [12]. This method is a combination of the Scale Invariant Local Ternary Operator (SILTP) and the Visual Background Extractor. SILTP was introduced by Liao *et al.* [16] to overcome some of the drawbacks of Local Binary Pattern and Local Ternary Pattern. There are three main advantages of the SILTP - it is computationally efficient, robust to local image noises within a range and finally, the scale invariance property of the operator makes it robust to illumination changes. However, SILTP by itself, fails when both the background and foreground objects have a similar texture information. ViBe is a non-parametric background subtraction algorithm. This algorithm utilizes the first frame of the video to initialize the background model based on the assumption that neighboring pixels share a similar temporal distribution. ViBe is based on the color or intensity information of pixels and hence, may fail when foreground objects have similar color as the background. The combination of SILTP and ViBe gives a background model which is robust to illumination changes. This model can be used in indoor, outdoor or night scenes - with little or no modifications. SILTP also allows for detection and efficient removal of shadows from the foreground.

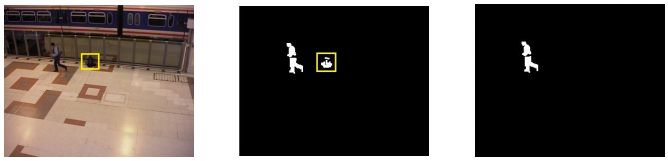
We put two learning rates of the model to create two background models - a long-term and a short-term model denoted as B_S and B_L respectively. B_L has a learning rate much slower than B_S. The stationary foreground objects are absorbed into the background faster in B_S than in B_L. We denote F_L and F_S for the binary images extracted from the long-term and short-term background models respectively. We denote 1 for the foreground pixel and 0 for the background pixel in the binary images. The difference between F_L and F_S is used to detect stationary foreground objects using a pixel-based Finite State Machine.

2.2. Pixel-based Finite State Machine

We introduce a novel, pixel-based Finite State Machine (PFSM) for an accurate detection of stationary foreground objects. The input of this PFSM is the binary values of F_L and F_S generated by B_L and B_S. We define the value of pixel i at time t as

$$S(i)(t) = F_S(i)(t)F_L(i)(t)$$

where $F_L(i)(t)$ and $F_S(i)(t)$ are the binary values of the i pixel in F_L and F_S frames generated at time t by B_L and B_S respectively. One example of the long term and short term frames generated by the two background models are shown in Figure 2 with a bounding box around an abandoned bag. At any instance of time t, each pixel i in the frame corresponds to a unique value $S(i)(t)\epsilon\{00, 01, 10, 11\}$:



(a) (b) (c)

Fig. 2: Example for various hypotheses of pixels - (a) is the original frame, (b) is the long-term model F_L, (c) is the short-term model F_S

- $S(i)(t) = 00$ indicates that pixel i is classified as background in both F_L and F_S.
- $S(i)(t) = 01$ indicates that pixel i is an uncovered background pixel that has been temporarily occluded by an object and then exposed in a recent frame.
- $S(i)(t) = 10$ indicates that pixel i is a foreground pixel in F_L and background in F_S which likely to be a static foreground pixel.
- $S(i)(t) = 11$ indicates that pixel i corresponds to foreground pixel in both F_L and F_S.

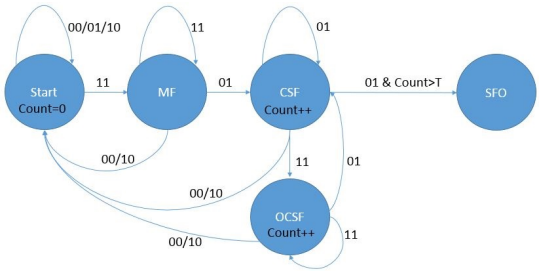


Fig. 3: State diagram of the pixel-based Finite State Machine. MF stands for Moving Foreground, CSF stands for Candidate Static Foreground, OCSF stands for Occluded Static Foreground, SFO stands for Static Foreground Object.

Based on the four inputs, the proposed PFSM has five states:

- Start
- MF(Moving Foreground)
- CSF (Candidate Static Foreground)
- OCSF (Occluded Static Static Foreground)
- SFO(Static Foreground Object)

The FSM designed, takes into account the *sequence of transitions* of the value of each pixel for identifying static foreground objects. Figure 3 shows the state diagram of the FSM for classifying an object as static foreground. The FSM is triggered by $S(i)(t) = 11$ which denotes a pixel constituting moving foreground and transitions to the state MF (Moving Foreground). The FSM will remain on this state as long $S(i)(t) = 11$. When a foreground object has been stationary for some time, the short-term background model absorbs this object into the background whereas it still shows up as foreground in the long-term model. This leads the pixels state $S(i)(t)$ to transition from 11 to 01. This transition triggers a state transition in the FSM. The pixels are now classified as CSF (Candidate Stationary Foreground). A count is initiated when the FSM first transitions to CSF state. This count keeps track of the duration for which the foreground object has been stationary. If $S(i)(t)$ changes from 01 to 11 there are two possible scenarios (i) the object is no longer stationary or (ii) the stationary object has been occluded by other moving foreground object. We are interested for case (ii). When $S(i)(t)$ transitions from 01 to 11, the FSM goes to state OCSF (Occluded Candidate Static Foreground). The count continues to be incremented in this case. If the static object has been occluded by a moving foreground object, then $S(i)(t)$ will return to 01 value when the occlusion is removed and the FSM will go back to CSF state. Otherwise the state goes back to the start state. If the object has been stationary for more than some specified time T, denoted by count becoming greater than T, then the object is classified as a stationary foreground object. This causes the transition of the FSM to state SFO (Stationary Foreground Object). This way, the FSM uses the pixels temporal transition information to detect stationary foreground object.

2.3. Object Detection and Classification

The FSM will detect any stationary entity in the scene. To reduce the false alarms arising from stationary people or other objects, we classify the stationary foreground objects into suspected objects which includes backpack, handbag, and luggage. We also detect person nearby the suspected detected object to check if the object is abandoned for a particular period of time. We use a method named SSD[15], for detecting the suspected objects and person in images using a single deep neural network. This method use multi-scale convolutional bounding box outputs attached to multiple feature maps at the top of the network. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. Moreover, the network combines predictions from multiple feature maps with different resolutions to naturally handle objects of various sizes. SSD is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network. This makes SSD easy to train and straightforward to integrate into systems that require a detection component. Some examples of person detection and suspected

objects(backpack) detection by SSD is shown in figure 4. In order to reduce the false positive alarm, we remove the stationary candidate object if the object is detected as person.

(a) (b)

Fig. 4: Example of (a) Person detection by SSD on an input image (b) Suspected object(backpack) detection by SSD on an input image

2.4. Abandonment Check

PETS2006 Workshop [17] has released a set of rules to classify an object as abandoned. We employ similar rules in our system to check for abandonment. We use two main rules to check if the luggage is attended by its owner or not (in which case, it is abandoned). The abandonment of luggage is defined spatially and temporally. These spatio-temporal rules employ a check on the results and remove any false positives arising from temporarily unattended luggage or just stationary luggage.

1) Temporal rule: The object is detected as abandoned object when it is left by its owner, and the luggage is not attended within time $T = 30$ seconds.

2) Spatial rule: The object is detected as abandoned object when it is left by owner and the distance between the owner and the object is greater than a predetermined distance $D_S = 3m$.

In the proposed PFSM, temporal rule is satisfied by putting $count = 30f$ frames, where f is the frames per seconds of the input video. To check the spatial rule, we create the suspected object centric circle with redius $D_S = 3\mu$, where μ denotes the scaling factor to convert pixel into real-world distance. We check the distance between the nearest detected person by SSD and the detected suspected object, denoted as S_i. We investigate whether $S_i \geq D_S$ if the object is abandoned. We will put an alarm when both these rules are satisfied.

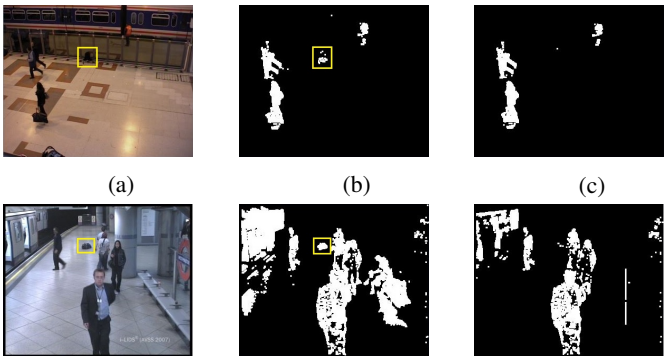
3. RESULTS AND ANALYSIS

3.1. Implementation Details

The proposed method is implemented in C/C++ with a 3.2 GHz Intel Core-i5 processor computer. We use OpenCV 2.4. Based on the study and research of [2], we also use the same learning rates for short and long term model. We use 10 and 100 frames to update background for short term and long term model respectively. For object classification, we use three objects on MS COCO 2014 and self collected dataset. The details are shown in table 1

Table 1: Details for Object detection and classification for SSD

Object	Number of Trining images	Number of Validation images	Iteration
Person	45174	21634	300K
Backpack	2910	100	120K
Handbag	5351	500	120K



(a) (b) (c)

(d) (e) (f)

Fig. 5: Results of original frame, long term and short term outputs by the proposed method: on PETS2006 (a-c), and on AVSS2007 dataset(d-f) respectively

3.2. Datasets and Results

We have tested this system on four standard, public datasets - PETS2006 [17], PETS2007 [18], AVSS2007 [19] and ABODA [20]. PETS2006 and PETS2007 contain videos shot from 4 camera angles. PETS2006 contains 7 scenes of abandoned luggage of varying difficulty. PETS2007 contains 2 scenes of temporarily abandoned luggage. AVSS2007 consists of three videos of varying difficulty - easy, medium and hard. ABODA consists of 11 videos of various indoor, outdoor, and night and day time scenes. Some examples of our output results are shown in Figure 5.

3.2.1. PETS2006

This dataset has been most widely used to benchmark abandoned object detection frameworks. It consists of 7 videos showing a railway platform from four different camera angles. Most of the previous papers ([13], [10], [21] , [2]) have only published their results for the camera angle 3 of the

Table 2: Detection results on PETS2006 dataset

Scene	Ours	[13]	[10]	[21]	[2]
S1	T	T	N/A	T	T
S2	T	T	T	T	T
S3	T	T	N/A	F	T
S4	T	N/A	N/A	T	T
S5	T	N/A	N/A	T	T
S6	T	N/A	T	T	T
S7	T	T	N/A	T	T

PETS2006 dataset. In contrast, we have evaluated our framework on four camera angles. It detects for all the cameras and scenes. Table 2 shows the comparison of results on camera angle 3 of this dataset. We classify a result as True (T) only when the warning is raised for an abandoned object without giving any false positives or detecting stationary people as abandoned objects. The work of Lin $et\ al.$ [2] shows the best results among the previously published works. However, they have only evaluated their method on the camera 3 which is a direct angle without many challenges. For the other camera angles, their open source code fails to detect the abandoned object in several instances and gives many false positives in others.

3.2.2. PETS2007

This dataset contains 2 videos with instances of abandoned luggage. Each scene is shot from four different camera angles. The luggage is temporarily unattended and retrieved by the owner shortly.As the object is abandoned for a very short duration, there should not be any alarm in either of the scenes. We tested our framework on each of the four angles of the two scenes. The framework doesn't raise an alarm in either of the scenes. On the contrary, [14] detects the luggage as an abandoned object and raises an alarm.

3.2.3. AVSS2007

This dataset consists of three videos of easy, medium and hard difficulty levels. The difficulty levels are classified on the basis of the location of abandoned object in the near zone (easy), mid zone (medium) and far zone (difficult). Our framework correctly detects the luggage in all the three scenes, however doesn't detect the luggage as abandoned in the easy scene. This is because in the easy scene the owner stands behind the luggage before walking away. Hence, parts of the luggage are detected as the owner's shadow. A broken foreground doesn't work on the PFSM and therefore, the object is not detected as an abandoned luggage. This is a drawback of the background modeling method that we have used.

Table 3: Detection results on ABODA dataset

Video	Scenario	Ours	[2]	[9]
V1	Outdoor	T	T	T
V2	Outdoor	T	T	T
V3	Outdoor	T	T	T
V4	Outdoor	T	T	T
V5	Night	T	F	T
V6	Illumination change	T	T	N/A
V7	Illumination change	T	F	N/A
V8	Illumination change	T	F	N/A
V9	Indoor	T	T	T
V10	Indoor	T	T	T
V11	Crowded scene	F	F	N/A

3.2.4. ABODA

The ABandoned Object DAtaset (ABODA) was created by Lin $et\ al.$ [2]. It consists of 11 videos depicting various indoor, outdoor, night scenes, and scenes with sudden illumination changes. Wahyono $et\ al.$ [9] use a reference background in their method. Hence, their framework cannot adjust to the illumination change and the subsequently the background change. Lin $et\ al.$ [2] cannot address significant illumination changes in their framework either. The advantage of using sViBe background model is that this method needs only one frame for initializing. So when the model detects an illumination change, the background model is reinitialized and quickly adjusts to the new conditions. This makes our framework robust to both sudden and gradual illumination changes. The results of the detection for this dataset are shown in table 3.

3.3. Special Features

3.3.1. Illumination Change

The advantage of using sViBe background model is that this method needs only one frame for initializing. So when the model detects an illumination change, the background model is reinitialized and quickly adjusts to the new conditions. This makes the proposed framework robust to both sudden and gradual illumination changes. The results of the detection for this dataset are shown in table 3. On the contrary, Wahyono $et\ al.$ [9] , Lin $et\ al.$ [2], Wahyono $et\ al.$ [9] and Lin $et\ al.$ [2] can not handle illumination changes.

3.3.2. Temporary Occlusion

In order to detect temporary occlusion, the proposed framework uses the OCSF(occluded static foreground) state in the FSM. The FSM keeps track of the static object even when it is occluded and hence the alarm is raised timely as it should have.

4. CONCLUSION

This paper proposes a framework for abandoned object detection by combining a PFSM model and SSD detector. Our framework outperforms the existing methods tested on the standard datasets. However, this framework still cannot handle extremely crowded scenes. In future, we plan to enhance our method to handle more crowded, wide-angle scenes and to a more robust model against fluctuating foreground pixels.

5. REFERENCES

[1] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Computer Vision and Pattern Recognition or CVPR*, 1999.

[2] K. Lin, S. C. Chen, C. S. Chen, D. T. Lin, and Y. P. Hung, "Abandoned Object Detection via Temporal Consistency Modeling and Back-Tracing Verification for Visual Surveillance," *IEEE Transactions on Information Forensics and Security*, 2015.

[3] R. Heras Evangelio and T. Sikora, "Complementary background models for the detection of static and moving objects in crowded environments," *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2011*, 2011.

[4] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, 2011.

[5] J. Dou and J. Li, "Moving object detection based on improved VIBE and graph cut optimization," *Optik - International Journal for Light and Electron Optics*, 2013.

[6] D. Jin, S. Zhu, X. Sun, Z. Liang, and G. Xu, "Fusing Canny operator with vibe algorithm for target detection," *Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016*, 2016.

[7] M. Van Droogenbroeck and O. Paquot, "Background subtraction: Experiments and improvements for ViBe," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012.

[8] Y. Yang, D. Han, J. Ding, and Y. Yang, "An improved ViBe for video moving object detection based on evidential reasoning," *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2016.

[9] Wahyono, A. Filonenko, and K. H. Jo, "Unattended object identification for intelligent surveillance system using sequence of dual background difference," *IEEE Transactions on Industrial Informatics*, 2016.

[10] L. Xiya, W. Jingling, and Z. Qin, "An Abandoned Object Detection System Based on Dual Background and Motion Analysis," *International Conference on Computer Science and Service System (CSSS)*, 2012.

[11] C. Cuevas, R. Martnez, D. Berjn, and N. Garca, "Detection of Stationary Foreground Objects Using Multiple Nonparametric Background-Foreground Models on a Finite State Machine," *IEEE Transactions on Image Processing*, 2017.

[12] H. Xu and F. Yu, "Improved compressive tracking in surveillance scenes," *Proceedings - 2013 7th International Conference on Image and Graphics, ICIG 2013*, 2013.

[13] J. Martinez del Rincón, J. Herrero-Jaraba, J. R. Gómez, and C. Orrite-Uruuela, "Automatic left luggage detection and tracking using multi-camera ukf," *9th PETS CVPR*, 2006.

[14] R. K. Tripathi, A. S. Jalal, and C. Bhatnagar, "A framework for abandoned object detection from video surveillance," *Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on*, 2013.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015.

[16] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikinen, and S. Z. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[17] "Performance evaluation of tracking and surveillance (pets) 2006 dataset," `http://www.cvg.reading.ac.uk/PETS2006/data.html`.

[18] "Performance evaluation of tracking and surveillance (pets) 2007 dataset," `http://www.cvg.reading.ac.uk/PETS2007/data.html`.

[19] "Advanced video and signal based surveillance (avss) 2007 dataset," `http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html`.

[20] "Abandoned object dataset (aboda)," `http://imp.iis.sinica.edu.tw/ABODA/index.html`.

[21] Y. Tian, R. S. Feris, H. Liu, A. Hampapur, and M. T. Sun, "Robust Detection of Abandoned and Removed Objects in Complex Surveillance Videos," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Systems*, 2011.