Noname manuscript No.
(will be inserted by the editor)

A Cellular Automata Based Approach to Track Salient Objects in Videos

Luca Crociani · Giuseppe Vizzari · Antonio Carrieri · Stefania Bandini

Received: date / Accepted: date

Abstract In this paper we present an algorithm to track the motion of a salient object using Cellular Automata. The overall work, taking inspiration from recent research on insect sensory motor system, investigates the application of non conventional computer vision approaches to evaluate their effectiveness in fulfilling this task. The proposed system employs the Sobel operator to individual frames, performing further elaborations through a CA, with the aim of detecting and characterizing moving entities within the field of view to support collision avoidance from the perspective of the viewer. The paper formally describes the adopted approach as well as its experimentation videos representing plausible situations.

Keywords Cellular Automata, Motion Detection, Video Analysis

1 Introduction

Motion detection and object tracking are both tasks of great interest in Computer Vision (CV), and they are used within a large number of very different application areas such as medical imaging, surveillance [31], and (of more recent interest) driver assistance [3]. The aim of this paper is to present a method for motion detection and characterization using Cellular Automata (CA): the defined approach, taking inspiration from recent research on insect sensory motor system, has the aim of detecting and characterizing moving entities to support collision avoidance from the perspective of the viewer. The present paper reports an ongoing work aimed at investigating a wider research challenge, that is, the possibility to transfer intuitions, approaches and concrete

L. Crociani
CSAI research center
University of Milano – Bicocca
Viale Sarca 336/U14, 20126, Milan (ITALY)
E-mail: luca.crociani@unimib.it

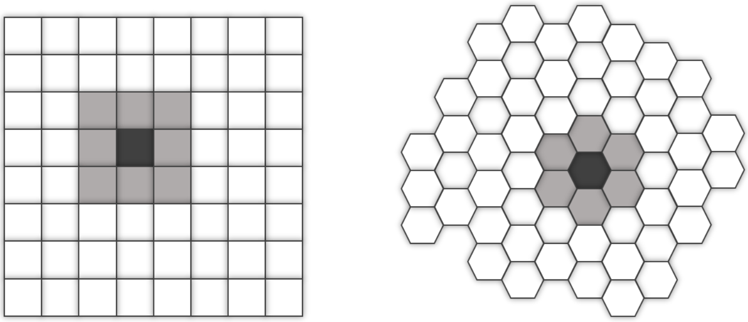
results from the field of insect sensory and motor system study to the area of autonomous robotics, in the vein of [2]. At this stage of the work, we are more interested in evaluating the potential effectiveness of the approach, but the reason for investigating non conventional computer vision approaches is the long term goal of defining techniques that can be implemented in efficient and cost-effective hardware solutions. Insects, in fact, by implementing relatively simple behaviors (also due to their relatively small brain and nervous system), achieve accurate and flexible overall sensory-motor skills such as tracking [1], visually guided collision avoidance [11], and they even compensate for the intrinsic intermittent and sporadic nature of chemical signals in pheromone-based search strategies [28, 29]. While studies on the latter phenomenon have already been used as a starting point for the development of so-called "sniffers" (i.e. robots that can track chemicals emitted by drugs, chemical leaks, explosives and mines), the potential for the transfer of concepts, mechanisms, approaches, to other practical applications to the design of autonomous vehicles is still to be explored.

Within this wider line of work, we identified in the edge detection, more specifically in the Sobel operator [26], a satisfactory algorithm that performs an efficient transformation of an image in its edge-based counterpart. This image transformation, leading to a gray-scale representation, can be easily translated in a cellular automaton configuration [30]. Although edge detection is a very consolidated specific field of computer vision techniques [5, 10, 21, 23], it is nonetheless possible to find some peculiarities that make it well suited to a cellular automaton approach. Likewise, intrinsic features of CA make them naturally suited to parallelization [27] and efficient hardware implementation [13], with the support of ad-hoc devices, they could bear the development and usage of a real time system. This work extends a conference paper [6] in particular by providing a mechanism performing a sort of commonsense spatial-temporal correlation based on a *discrimination buffer*, following the CA-based element of the overall processing pipeline reducing the tracking errors.

A more thorough discussion of relevant related works will be given in the following section, while a formal description of the proposed approach will be provided in Sect. 3, together with a running example showing results of the different phases of the overall work flow. An additional example and discussion of the achieved results and current state of the work will be described in Sect. 4. Conclusions and future developments will end the paper.

2 Related Works

The algorithm proposed in this paper involves CA which is a mathematical framework widely applied for the modeling of complex systems, ranging from physical systems [9], to systems in which the collective human behavior is the object of study, both at a small scale (e.g., pedestrian movement [4]) as well as in urban scale [25]. In CA models space and time are discrete, and the



(a) (b)

Fig. 1 Example of (a) rectangular and (b) hexagonal grid of 2-dimensional CA. Highlighted cells indicate possible neighborhood structures.

environment generally consists of a regular uniform lattice. Figure 1 shows the two most used grid types to define a CA. Each site of the grid is also denoted as "cell". The latter are described by a specific state, which is synchronously modified by a local update rule considering on the state of its neighbors. The neighborhood of a cell can be modeled in various ways, but it generally includes the cell itself and other cells reachable by means of an adjacency relation: in case of a 2-dimensional rectangular grid, for example, the Moore neighborhood includes the eight surrounding cells of the subject one, while the Von Neumann neighborhood does not consider diagonal neighbors. While the discrete nature of the CA, together with the description of simple local rules, provides a rather simple approach, the iteration of the local rules over the time-steps is able to generate highly complex and unpredictable future states of the systems.

CA approaches have already been used for different tasks within the computer vision area. Popovici and Popovici [20], for example, proposed a comparison of two CA based algorithms with other approaches from the literature for noise removal and border detection, highlighting the qualities of results achieved by the works employing CAs.

Regarding the edge detection task, an earlier approach based on CA was proposed in [8]. The authors introduce an ad-hoc neighborhood structure, so-called semi-neighborhood, which is used in the iterative computations of the CA states. Starting with a pre-processed input image with a method to extract the orientation information, results shown in the paper highlight good performances of the algorithm in detecting edges. A more recent work for the processing of edge detection is published in [19]. The paper proposes a Cellular Automaton that is able to outperform other literature algorithms both in terms of effectiveness and computation times.

Additional works for the processing of images by means of CA can be found. For example, [24] describes and evaluates an algorithm for the segmentation of medical images. Moreover, in [16] the authors proposed a CA based algorithm

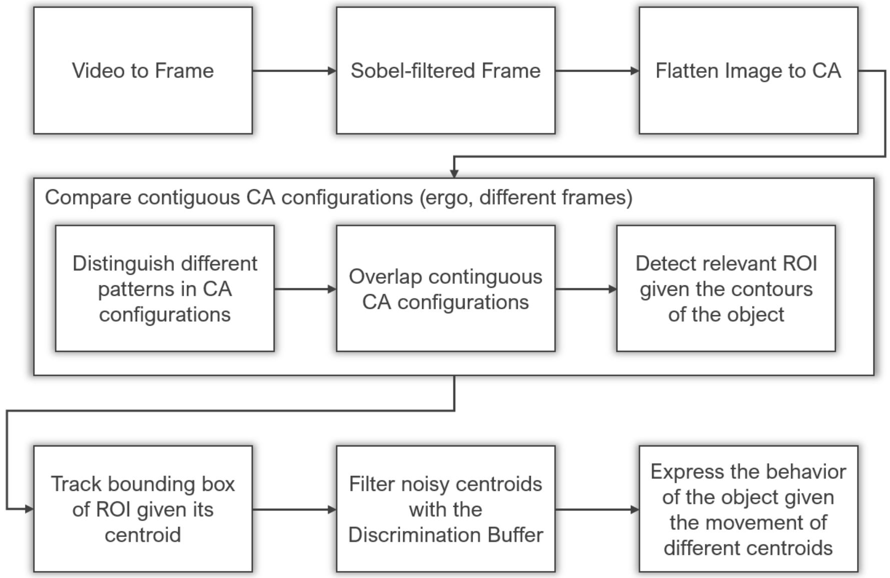


Fig. 2 The overall pipeline of the proposed approach for CA-based motion detection and characterization.

to increase the resolution of an image while, at the same time, preserving its edges and overall quality.

Even though CAs are not currently among the most widely adopted approaches for motion detection, recent works aimed at recognizing saliency in pictures can also be found. A relevant example is provided by the work in [22] which employs a stochastic CA approach to detect the most salient object in pictures. This work has been well received by the computer vision community due to its effectiveness and efficiency. Saliency detection analysis with CA, in fact, was also later investigated in [14]. In general, as suggested in the introduction, besides the inspiration to the insect sensory-motor system, we also considered the CA approach due to the possibility to translate CA lattice and rules from a software implementation to a hardware specific one, maybe employing a field-programmable gate array (FPGA), to take full advantage of the inherent parallelism of the CAs, in the vein of [17].

3 The Proposed Approach

Our approach and the associated work flow implies several sequential steps in order to process a frame-by-frame object movement. After a pre-processing of the input frame required to clean and standardize the information, the central components of the pipeline are performed with the proposed CA approach, as shown in Fig. 2.

-1	0	1
-2	0	2
-1	0	1

(a)

1	2	1
0	0	0
-1	-2	-1

(b)

Fig. 3 (a) Kernel matrix used along the x-axis (G_x). (b) is used for the y-axis (G_y).

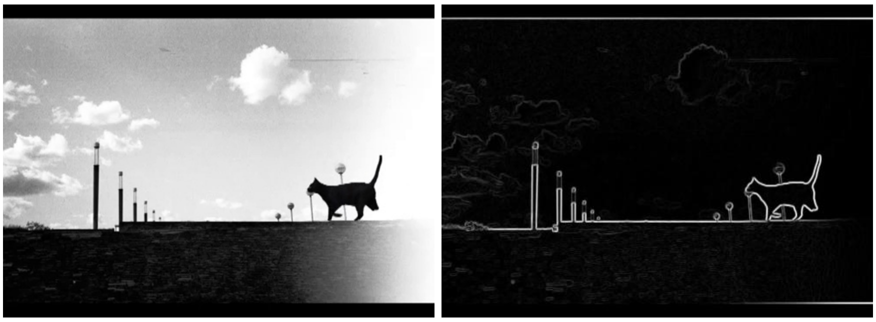
To improve the understanding of the algorithm, we will report the results of the relevant steps of the process with an example application. For this purpose, we use a simple video[1] with a static camera, whose frame resolution is 496×360 pixels: the proposed approach does not depend on a particular resolution, and it provides results analogous to the ones we will show and discuss later on for resolutions not smaller than few hundreds of pixels per side, with increased computational costs with the growth in resolution. The background is static, unless for artifacts produced by the video compression and for slight changes in the illumination. The video shows a cat entering the scene from the right side and moving towards the other end.

3.1 Pre-processing of the Frame

To transform an image into an instance of a CA, every frame of a video will be filtered using the Sobel operator. The operator applies two 3×3 kernels to the original image in order to calculate approximations of the derivatives in the horizontal and vertical directions (see Fig. 3). The gradient $\mathbf{G}$ of the edge is calculated as $G = \sqrt{G_x^2 + G_y^2}$. This filter provides a transformation of the input which helps the process of salient motion recognition: by applying this filter the colors are going to be removed, and only edges in grey scale will be highlighted. The edges are basically areas where contrast intensity $\gamma \in \mathit{\Gamma}$ is strong. Filtering an image with this operator, provides a new image which will be used to initialize a CA lattice.

Reasons behind the usage of Sobel operator rather than other edge detectors can be found in its simplicity. While other edge detectors (e.g. Canny edge detector) can provide a cleaner and more complete output image, they generally imply a higher number of steps to achieve the final result (see, e.g., the discussion in [16]). In addition, the workflow of the Sobel operator is lighter in terms of computations to perform, and it is also able to detect edges on a regular basis even in particularly hard cases (e.g. having an object that does

[1] https://www.youtube.com/watch?v=HDb9StNG8_Q



(a) (b)

Fig. 4 A sample frame of the video of (a) the moving cat and (b) the result achieved by applying the Sobel filter.

not have well defined contours). A more thorough evaluation of edge detection algorithms can be found in [18], but these characteristics motivate our choice of the Sobel operator for the first step of the workflow.

In Figure 4 the result of the Sobel operator is shown in the sample video: given the input frame in grey scale, it returns a filtered image showing the detected edges based on changes of contrasts. As it can be seen in Figure 4(b), the operator was particularly effective in detecting the edges of the cat.

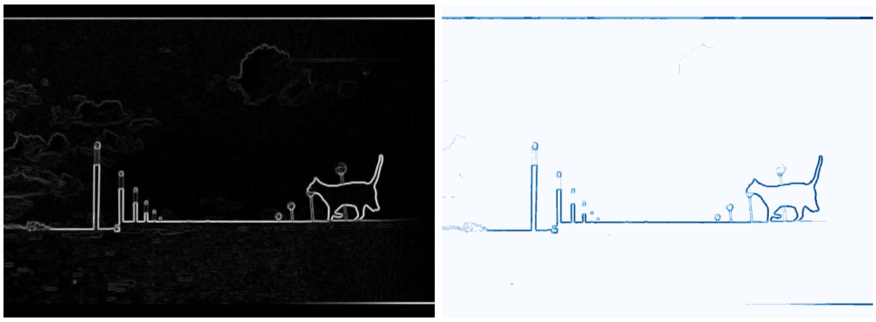
3.2 CA Initialization

To compute a discrete and regular input to be processed by the rules of the CA, the grid of contrasts Γ output by the Sobel filter is grouped in clusters in this phase of the algorithm, discretized to initialize the lattice of the CA. The number of clusters is determined according to the content of the processed video with the aim of preserving the possibility to discriminate edges but also to keep limited the processing time. So once clustered, there will be a finite set of states $S = \{0, ..., k\}$ every cell can assume.

Therefore defining a frame $F^t = \{p^t_0, p^t_1, ..., p^t_{(n*m)-2}, p^t_{(n*m)-1}\}$, where n is the number of pixels on the x axis and m the number of pixels on the y axis, as the t_{th} frame in a video $V = \{F^0, F^1, ..., F^{max(t)}\}$, the flattening process will follow this method:

$$S(c^t_i) = \begin{cases} k-1, & \text{if } min(\gamma_{K^n}) \leq \gamma_{p^t_i} \leq max(\gamma_{K^n}) \\ \vdots & \\ 1, & \text{if } 0 < min(\gamma_{K^1}) \leq \gamma_{p^t_i} \leq max(\gamma_{K^1}) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $S(c^t_i)$ denotes the state of cell c_i of the CA lattice (for $i \in \{0, \ldots, (n*m)-1\}$) at time t.



(a) (b)

Fig. 5 (a) The Sobel-filtered frame and (b) the output grid given for the CA initialization.

In our experiments, for the discretization we adopted a 8 bit encoding for the CA, therefore with $k = 256$; we also decided to set the intervals γ_{K^i} to achieve a uniform partition of the space of possible gradient levels, although in theory other forms of (also non-linear) discretization are possible (for instance to better delineate elements within the frame).

At the end of this process there will be a fully initialized lattice L describing the input to the Cellular Automaton, with cells having a state $s \in [0, k]$.

Figure 5 shows this additional processing of the input with the Sobel-filtered frame previously output by the first step of the workflow (the different states of the new grid are represented in blue scale). The flattening of the input is visible, and it eases the subsequent steps dedicated to the comparison of frames. This step overall aims at removing superfluous edges that should not be considered for further evaluation.

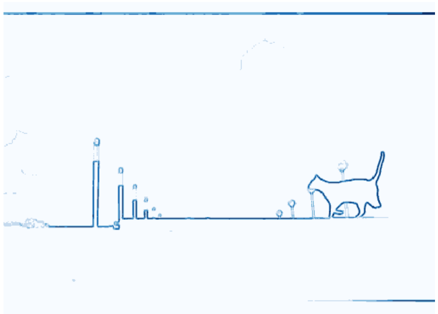
3.3 Frame Comparison

After having the grid of flattened values that potentially identifies different objects in the scene, a process of frames comparison to characterize the movement within the considered video can be performed. In order to do this we will use two different lattices which are contiguous in time, respectively denoted as $L(F^t)$ and $L(F^{t+1})$. The two grids are overlapped in the process to retrieve uncommon cells according to their position. As a result, a new lattice $\Lambda(L(F^t), L(F^{t+1}))$ is produced according to this method:

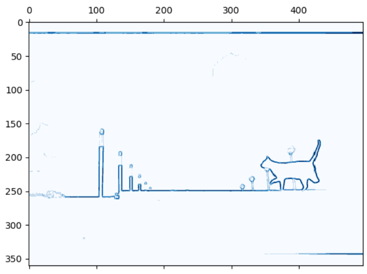
$$S(c_i^{t,t+1}) = \begin{cases} 1, & \text{if } S(c_i^t) \neq S(c_i^{t+1}) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $S(c_i^{t,t+1})$ represents the state of the $i-th$ cell of the above introduced lattice $\Lambda(L(F^t), L(F^{t+1}))$ (for $i \in \{0, \ldots, (n * m) - 1\}$).

$\Lambda(L(F^t), L(F^{t+1}))$, hence, simply describes different pixels in the two input frames and it represents the base information for the movement detection



(a)



(b)

(c)

Fig. 6 (a–b) The CA configuration of two analyzed frames (for making more intelligible the difference among frames non consecutive ones have been selected) and (c) the resulting grid $\Lambda(L(F^{104}, L(F^{106}))$.

process. Given that both $L(F^t), L(F^{t+1})$ were output of the Sobel operator, this new lattice presents edges that were present at time t and that changed at time $t+1$: as it is shown in Figure 6, the resulting grid does not contain edges related to the background, but it includes the ones describing the moving object related to both time t and $t+1$.

In order to determine the so-called region of interest (ROI) and to finally characterize the salient motion in the two distinct frames, we must firstly separate the pixels that describe movement in the two frames. In particular, we have to exclude cells that do not match their state value when compared to $L(F^t)$ and when compared to $L(F^{t+1})$ from the lattice $\Lambda(L(F^t), L(F^{t+1}))$. Therefore, this process will bring to two new different lattices $ROI(L(F^t))$ and $ROI(L(F^{t+1}))$. Respectively, their cell states will be set according to the following equations:

$$S(c_i^{ROI(L(F^t))}) = S(c_i^t) * S(c_i^{t,t+1}) \tag{3}$$

$$S(c_i^{ROI(L(F^{t+1}))}) = S(c_i^{t+1}) * S(c_i^{t,t+1}) \tag{4}$$

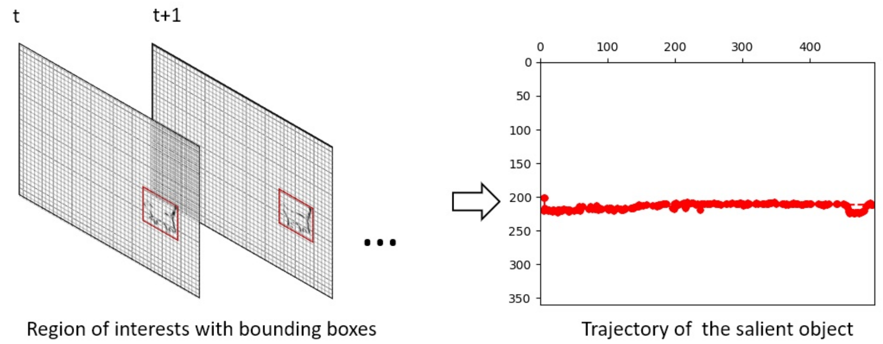


Fig. 7 Tracking the salient object using bounding boxes to enclose the detected ROIs. Each point of the diagram in the right represents the centroid of the bounding box of a given frame.

After this step of the process, the pixels associated to the moving salient object are identified and the motion can be finally tracked.

3.4 Building a Bounding Box around the Salient Object

The process of tracking is performed by estimating the position of the set of moving pixels computed in each frame. To do this, we construct a bounding box around the region of interest and we consider its centroid as the position of the moving object in the current frame. A trajectory of all of the bounding boxes will show the approximate behavior of the moving object in the whole video. This process is illustrated in Figure 7.

3.5 Introducing a Discrimination Buffer

To allow the evaluation of the effectiveness of the whole pipeline, and to compute the results which have been already presented in the previous Section, the algorithm has been implemented using Python programming language and SciPy (ndimage)[2] library for the Sobel filtering part along with OpenCV[3] for several tasks on the video processing.

At this point of the pipeline, the algorithm can already be used to provide the tracking of a single object in a video. Figure 8(a) shows the trajectory of the cat in the example video. While some errors are particularly visible, the diagram allows to understand the actual movement of the animal: in an initial part of the video (frames *1* to *49*) there is no motion (the cat has not yet entered the screen) and consequently nothing is detected; starting at frame

[2] `https://docs.scipy.org/doc/scipy/reference/ndimage.html`

[3] `https://opencv.org/`

50 and until frame *268* the system detects an object moving at a relatively constant speed from the right side of the frame to the left side. Finally, the sequence of frames between *269* and *293* depict the background since the cat has exited from the right side of the screen, and the system does not report any movement.

A few imperfections in this result are given by the fact that the movement of the cat's tail was fairly unpredictable, leading to sudden changes of the bounding box shape: while the cat was regularly moving, sometimes the bounding box kept almost the same position in contiguous frames. In addition, a couple of frames around coordinates (300,150) were generated by noise appeared in the video that was also identified as a moving element. This has caused rather big bounding boxes for those frames, whose centroid actually led to the shifted points in the diagram. On the other hand, such simple approach overall led to a quite acceptable result in this case. For this reason, we introduce an additional step in the pipeline of the algorithm that is aimed at refining the final output and providing a cleaner trajectory of the moving salient object.

This step is executed with the use of a so-called *discrimination buffer*, where centroids and areas of the bounding boxes in the last k frames (k is a parameter of the algorithm) are temporarily stored, to evaluate the acceptance of the next one. Two criteria have been defined to filter the errors highlighted with the previous discussion. Firstly, the next centroid of a bounding box will not be accepted if the size of the bounding box is significantly different than the average size of the bounding boxes observed in the discrimination buffer. Another parameter β is introduced to quantify this concept, basically describing the percentage of variation of the bounding box size between subsequent frames.

The second constraint introduced to accept a new centroid, analyzes the velocity of the object in the two dimensions described in the video frame. In particular, a new centroid c_t related to the frame t will be accepted only if the Euclidean distance with its direct predecessor c_{t-1} is lower than the average distance between the centroids inside the discrimination buffer multiplied by a third parameter γ.

Figure 8(b) shows the resulting trajectory achieved with the proposed extension using the discrimination buffer. The output is sensibly cleaner than what was resulting from the baseline version, with just one exception that happened at the beginning of the video and led to a couple of points around coordinates *(110,210)*: at the really beginning of the video, the cat was still out of sight from the camera view but some noise due to video compression was detected as salient. The video, in fact, presents some issues in terms of compression artifacts, leading to a slight change of colors of pixels in certain frames. For the moment, a coarse compression as the one applied in this video still represents an issue for the proposed approach, and for this reason the authors recommend to apply to stabilized videos with slightly higher resolutions.

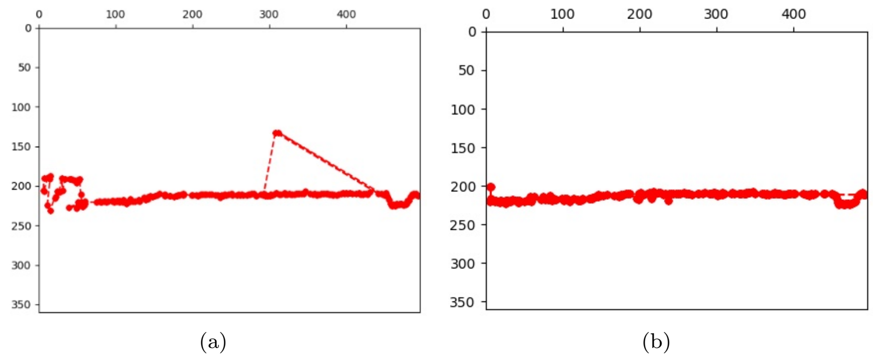


Fig. 8 (a) The achieved trajectory of the moving cat without applying the discrimination buffer and (b) with the proposed extension of the algorithm.

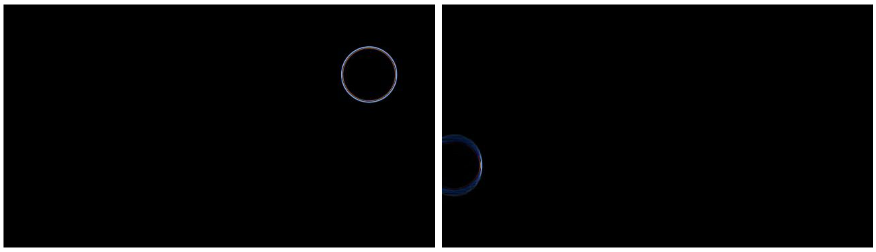
4 Application with a Different Video

In this Section, we will analyze the behavior of the algorithm with another case of application; in particular, the new experiment is performed with video[4] of a ball bouncing on the screen from the left to the right side. The video was chosen to evaluate the adequacy of the approach to situations in which the object to be tracked presented significant changes in the velocity of the movement within the frame: the ball is in fact quite slow in elevated positions, whereas it is fast when approaching the floor, where it also suddenly changes the direction of movement.

Figure 9 shows the detected edges of the ball in two different frames, after applying the Sobel operator. Figure 9(b) shows one issue related to this video: both while bouncing and when the ball reaches the borders of the frame, its shape gets deformed, making the size of the bounding box quite dynamic also in this example.

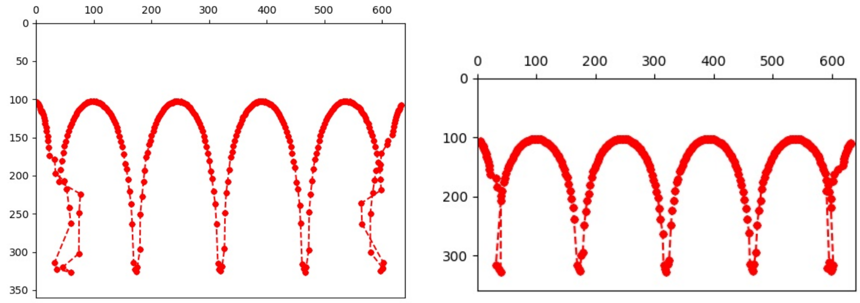
Moreover, the simulated gravity makes the object to constantly change its velocity along the y-axis (although for the x-axis is essentially constant), even with relatively significant displacements within the frame. Given that the video compression in this case does not provide sudden changes of colors in the black background, results for this scenario are quite satisfactory even without the use of the discrimination buffer, as shown in Figure 10(a). In all frames of the video, in fact, the ball is always detected as the salient object. Actually, the number of frames showing discrepancies between the expected bounding box position and the output one is 5 out of 295, the errors made in the estimated trajectory for those frames is very small and its location is limited to the sides of the frame (see the points at the borders of the picture). As introduced before, in fact, in those frames the ball speed is rather high and its edges become blurry. This makes the Sobel filter face some difficulties

[4] `https://www.youtube.com/watch?v=SW3rvS3wLqg` from which we digitally removed the "Ball" text.



(a) (b)

Fig. 9 (a) A frame taken from the video of a ball bouncing on screen from left to right. (b) A frame where the left edge of the ball is not completely on screen (the ball is in the lower right part of the frame and it is much less visible than in the first frame).



(a) (b)

Fig. 10 (a) Trajectory of the object achieved with the baseline version of the algorithm, compared to (b) trajectory output by the algorithm by using the discrimination buffer. Notice that aspect ratios are different to highlight points in which the discrimination buffer has a significant impact.

in producing well defined edges. Only the right edge of the object is therefore detected and the bounding box built around it makes its centroid shifted along the two axes.

By applying the proposed extension with the discrimination buffer, on the other hand, these bad features are skipped due to the filtering of the related centroids. Figure 10(b) shows the final output of the algorithm, highlighting a very clean and regular trajectory that perfectly describes the behavior of the ball during the video.

5 Conclusions and future works

The paper has discussed the first results of a larger effort aimed at investigating the possibility to transfer intuitions, approaches and concrete results from the field of insect sensory and motor system study to the area of autonomous robotics. The present results show that Cellular Automata can represent useful blocks within a larger work-flow for the processing of videos, in particular with the aim of detecting and characterizing motion within the analyzed frame. The two proposed experiments for the evaluation of the algorithm have shown a good effectiveness and are promising for future extensions and refinements of the approach. In particular, some issues outcome from the first video were due to compression artifacts, but also by the fact that the movement of the tail of the cat was fairly unpredictable and led to continuous changes of the bounding box shape. It is a matter of fact that this pointed out different movement directions between the cat and its own tail. On top of that, this method does not consider the problem of object classification, meaning that it does not consider the case of having more objects moving in the same frame yet. Thus, the approach is still far from being generally applicable in the real world, for the task of salient object motion detection.

On the other hand, considering some physical constraints characterizing the typically observed objects with the concept of "discrimination buffer" have already led to significantly improvements of the previously achieved results with a baseline version of this algorithm. In addition, with respect to the implementation aspect, due to the high level of parallelization of CA, we would like to focus our work on the classification of moving elements in an image, in order to process more objects within the CA. Regarding the classification problem, the greatest challenge is to reduce complexity in terms of computations.

The overall aim of this project is to transfer knowledge from the field of insect sensory and motor system study to the area of autonomous robotics. Existing studies dealing both with nonbound [7] and bound insects [12] show that collision avoidance plays a very important role within the more general sensory-motor integration of insects.

A work that could be likely associated to this project and that could be taken as inspiration for future implementations is, for example, [15] which shows a bio-inspired vehicle collision detection system using the neural network of a locust. While this work uses effectively cameras to process videos, our project would aim to do this with a CA abstracting the photo-receptor layer of the locust using a CA lattice.

References

1. Ando, N., Kanzaki, R.: A simple behaviour provides accuracy and flexibility in odour plume tracking – the robotic control of sensory-motor coupling in silkmoths. Journal of Experimental Biology **218**(23), 3845–3854 (2015).
2. Ando, N., Kanzaki, R.: Using insects to drive mobile robots—hybrid robots bridge the gap between biological and artificial systems. Arthropod Structure & Development **46**(5), 723 – 735 (2017)
3. Avidan, S.: Support vector tracking. IEEE transactions on pattern analysis and machine intelligence **26**(8), 1064–1072 (2004)
4. Bandini, S., Crociani, L., Vizzari, G.: An approach for managing heterogeneous speed profiles in cellular automata pedestrian models. J. Cellular Automata **12**(5), 401–421 (2017).

5. Canny, J.: A computational approach to edge detection. In: Readings in Computer Vision, pp. 184–203. Elsevier (1987)
6. Carrieri, A., Crociani, L., Vizzari, G., Bandini, S.: Motion detection and characterization in videos with cellular automata. In: Cellular Automata - 13th International Conference on Cellular Automata for Research and Industry, ACRI 2018 *Lecture Notes in Computer Science*, vol. 11115, pp. 102–111. Springer (2018).
7. Chan, R.W., Gabbiani, F.: Collision-avoidance behaviors of minimally restrained flying locusts to looming stimuli. Journal of Experimental Biology **216**(4), 641–655 (2013)
8. Chang, C.L., Zhang, Y.j., Gdong, Y.Y.: Cellular automata for edge detection of images. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826), vol. 6, pp. 3830–3834. IEEE (2004)
9. Chopard, B.: Cellular Automata Modeling of Physical Systems, pp. 407–433. Springer New York, New York, NY (2012).
10. Deriche, R.: Optimal edge detection using recursive filtering. International Journal of Computer Vision **2**, 167–187 (1987)
11. Fotowat, H., Gabbiani, F.: Collision detection as a model for sensory-motor integration. Annual Review of Neuroscience **34**(1), 1–19 (2011).
12. Fotowat, H., Gabbiani, F.: Collision detection as a model for sensory-motor integration. Annual review of neuroscience **34**, 1–19 (2011)
13. Georgoudas, I., Kyriakos, P., Sirakoulis, G., Andreadis, I.: An fpga implemented cellular automaton crowd evacuation model inspired by the electrostatic-induced potential fields. Microprocessors and Microsystems **34**(7), 285 – 300 (2010)
14. Guo, J., Ren, T., Huang, L., Liu, X., Cheng, M.M., Wu, G.: Video salient object detection via cross-frame cellular automata. In: Multimedia and Expo (ICME), 2017 IEEE International Conference on, pp. 325–330. IEEE (2017)
15. Hartbauer, M.: Simplified bionic solutions: a simple bio-inspired vehicle collision detection system. Bioinspiration & biomimetics **12**(2), 026007 (2017)
16. Ioannidis, K., Andreadis, I., Sirakoulis, G.C.: An edge preserving image resizing method based on cellular automata. In: Cellular Automata, pp. 375–384. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
17. Kalogeropoulos, G., Sirakoulis, G.C., Karafyllidis, I.: Cellular automata on FPGA for real-time urban traffic signals control. The Journal of Supercomputing **65**(2), 664–681 (2013).
18. Katiyar, S., Arun, P.: Comparative analysis of common edge detection techniques in context of object extraction. arXiv preprint arXiv:1405.6132 (2014)
19. Kumar, T., Sahoo, G.: A novel method of edge detection using cellular automata. International Journal of Computer Applications **9**(4), 38–44 (2010)
20. Popovici, A., Popovici, D.: Cellular automata in image processing. In: Fifteenth International Symposium on Mathematical Theory of Networks and Systems. Citeseer (2002)
21. Prewitt, J.M.: Object enhancement and extraction. Picture processing and Psychopictorics **10**(1), 15–19 (1970)
22. Qin, Y., Lu, H., Xu, Y., Wang, H.: Saliency detection via cellular automata. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, pp. 110–119. IEEE (2015)
23. Roberts, L.G.: Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology (1963)
24. Rundo, L., Militello, C., Russo, G., Pisciotta, P., Valastro, L.M., Sabini, M.G., Vitabile, S., Gilardi, M.C., Mauri, G.: Neuro-radiosurgery treatments: MRI brain tumor seeded image segmentation based on a cellular automata model. In: Cellular Automata - 12th International Conference on Cellular Automata for Research and Industry, ACRI 2016. *Lecture Notes in Computer Science*, vol. 9863, pp. 323–333. Springer (2016)
25. Santé, I., García, A.M., Miranda, D., Crecente, R.: Cellular automata models for the simulation of real-world urban processes: A review and analysis. Landscape and Urban Planning **96**(2), 108 – 122 (2010).
26. Sobel, I.: An isotropic 3× 3 image gradient operator. Machine vision for three-dimensional scenes pp. 376–379 (1990)
27. Toffoli, T., Margolus, N.: Cellular automata machines: a new environment for modeling. MIT press (1987)

28. Vergassola, M., Villermaux, E., Shraiman, B.I.: ‘Infotaxis’ as a strategy for searching without gradients. Nature **445**, 406 (2007).
29. Voges, N., Chaffiol, A., Lucas, P., Martinez, D.: Reactive searching and infotaxis in odor source localization. PLOS Computational Biology **10**(10), 1–13 (2014).
30. Wolfram, S.: Cellular automata as models of complexity. Nature **311**(5985), 419–424 (1984).
31. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. Acm computing surveys (CSUR) **38**(4), 13 (2006)