Motion Detection and Characterization in Videos with Cellular Automata

Antonio Carrieri, Luca Crociani, Giuseppe Vizzari(✉), and Stefania Bandini

Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo), Universitá degli studi di Milano Bicocca, Milan, Italy
a.carrieri4@campus.unimib.it
{luca.crociani,giuseppe.vizzari,bandini}@disco.unimib.it

Abstract. In this paper we present a method for motion detection and characterization using Cellular Automata. The original approach employs results of the application of the Sobel operator to individual frames, that are translated to CA configurations that are processed with the aim of detecting and characterizing moving entities 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). They are part of studies, for example, in medical imaging, surveillance methods [22] and (of more recent interest) driver assistance [2] and many other applications. The aim of this paper is to present a method for motion detection and characterization using Cellular Automata. The approach has the aim of detecting and characterizing moving entities to support collision avoidance from the perspective of the viewer.

In order to pursue this goal we identified in the edge detection, more specifically in the Sobel operator [19], an algorithm that performs an efficient transformation of an image in its edge-based counterpart with satisfactory effectiveness. This image transformation, leading to a gray-scale representation, can be easily translated in a cellular automaton configuration [21]. Considering that edge detection [3,5,15,17] is a very specific field of computer vision technique, it is nonetheless possible to find some peculiarities that fit well in the cellular automaton approach.

Likewise, intrinsic features of cellular automata make them naturally suited to parallelization [20] and efficient hardware implementation [7], with the support of ad-hoc devices, they could bear the development and usage of a real time system. We will now briefly discuss most relevant related works to this research, then the approach will be introduced. Discussion of achieved results and future research directions will end the paper.

G. Mauri et al. (Eds.): ACRI 2018, LNCS 11115, pp. 102–111, 2018.
https://doi.org/10.1007/978-3-319-99813-8_9

2 Related Works

Even though works related to motion detection using Sobel operator and CA are not present in the literature, Cellular Automata have recently been used for saliency detection [16]: the cited work, employing a stochastic CA approach, has been well received by the CV community being characterized, at the same time, by a good effectiveness and high efficiency, and it actually generated interest and further researches. Saliency detection analysis with CA, in fact, was later also investigated in [8], which also characterized it as one of the most relevant steps of the process of motion detection. CA approaches had been earlier used for other CV tasks, in particular to process edge detection [12,14] and to perform resizing operation preserving edges (and therefore quality of the image) [10], but also for segmentation of medical images [18].

3 The Introduced CA Approach

Our approach and the associated work-flow implies several steps in order to process a frame-by-frame object movement, as shown in Fig. 1. It involves Cellular Automata (CA) which is a mathematical idealization of physical systems in which space and time are discrete. It consists of a regular uniform lattice where, in each site, there is a discrete variable called "cell". Each individual cell is in a specific state and changes synchronously depending on the state of its neighbors, given a local update rule. The neighborhood at a certain site is typically taken to be the site itself and its immediate adjacent sites.

3.1 From a Frame to a Sobel-Filtered Frame

To transform an image into an instance of a CA, every frame of a video will be filtered using the Sobel operator. The latter applies two 3×3 kernels to the original image in order to calculate approximations of the derivatives, horizontal-axiswise and vertical-axiswise (see Fig. 2). Therefore the gradient **G** of the edge will be $G = \sqrt{G_x^2 + G_y^2}$. Because of its approximated nature, this filter helps in the process of discretization of an image. Applying this filter, colors are going to be removed, highlighting only edges in scales of gray. Edges are basically areas where contrast intensity $\gamma \in \Gamma$ is strong. Filtering an image with this operator, provides a new image which will be used to initialize a CA lattice.

The main reason for the usage of Sobel operator rather than other edge detectors can be found in the simplicity of the related algorithm. While other edge detectors (e.g. Canny edge detector) imply various steps to process the image and achieve its edge-based counterpart, as explained in [10], the Sobel operator edge detection method instead implies a shorter number of steps that are part of a much simpler algorithm.

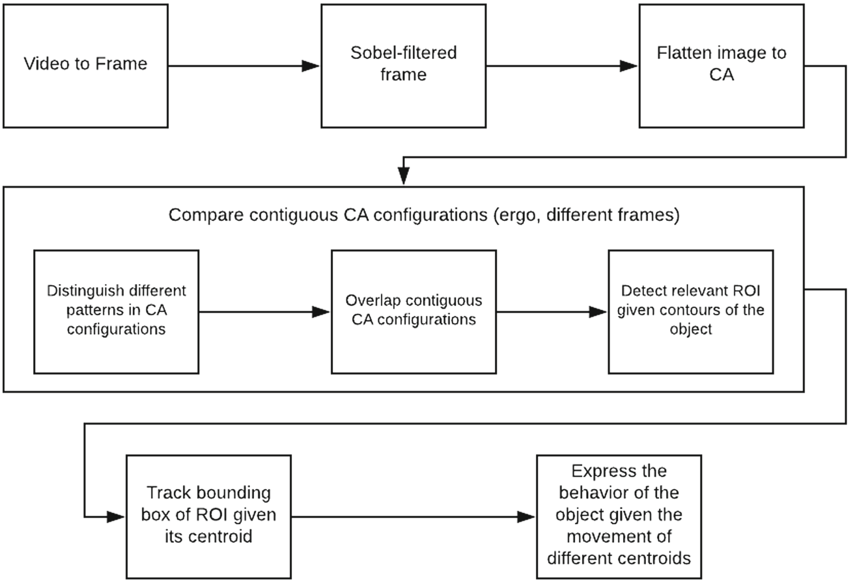
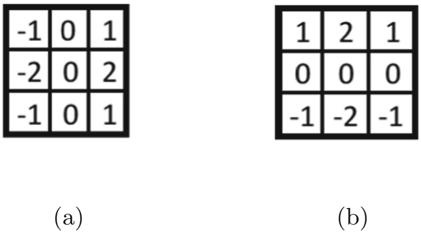

Video to Frame
Sobel-filtered frame
Flatten image to CA
Compare contiguous CA configurations (ergo, different frames)
Distinguish different patterns in CA configurations
Overlap contiguous CA configurations
Detect relevant ROI given contours of the object
Track bounding box of ROI given its centroid
Express the behavior of the object given the movement of different centroids

Fig. 1. The overall pipeline of the proposed approach for CA-based motion detection and characterization.



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

(a)

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

(b)

Fig. 2. (a) Matrix used on x axis (G_x); (b) Matrix used on y axis (G_y).

3.2 CA Initialization

Due to the intrinsic discrete nature of a CA, the actual set of contrasts Γ, processed by the Sobel filter, needs to be discretized in clusters. The cardinality of these clusters will be set as the highest value that a cell $c_i \in C$, where $i = 1 \dots |C|$, in a lattice L can assume. 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 $S = \{0, \dots, K\}$ every cell can assume.

Therefore defining a frame $F^t = \{p_0^t, p_1^t, \dots, p_{(n*m)-1}^t, p_{n*m}^t\}$, 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 $V = \{F^0, F^1, \dots, F^{max(t)}\}$, the flattening process will follow this method:

$$S(c_i^t) = \begin{cases} k-1, & \text{if } min(\gamma_{K^n}) \leq \gamma_{p_i^t} \leq max(\gamma_{K^n}) \\ \vdots & \\ 1, & \text{if } 0 < min(\gamma_{K^1}) \leq \gamma_{p_i^t} \leq max(\gamma_{K^1}) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

At the end of this process there will be a fully initialized lattice L with cells assuming up to k different states which will be associated to a sobel-filtered video frame.

3.3 Frames Comparison

Having the lattice set, a process of frames comparison to elaborate movement within the considered video will start. In order to do this, we will use 2 different, but contiguous in time, lattices $L(F^t)$ and $L(F^{t+1})$; they will be overlapped to retrieve uncommon cells according to their position. As a result a new lattice $\Lambda(L(F^t), L(F^{t+1}))$ will be produced according to this method:

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

In other words, lattice $\Lambda(L(F^t), L(F^{t+1}))$ will essentially show different pixels from each frame, which intuitively represent the focus of the movement detection process. More precisely, this new lattice presents edges that were present at time t and that changed at time $t+1$: it therefore includes edge pixels of both time t and $t+1$.

In order to determine more precisely the so called region of interest (ROI) of the distinct frames, we have to separate this information, to be then analyzed to characterize movement. More precisely, we would have to exclude from the lattice $\Lambda(L(F^t), L(F^{t+1}))$ cells that do not match their state value when compared to $L(F^t)$ cells and when compared to $L(F^{t+1})$ cells. 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 this method:

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

and

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

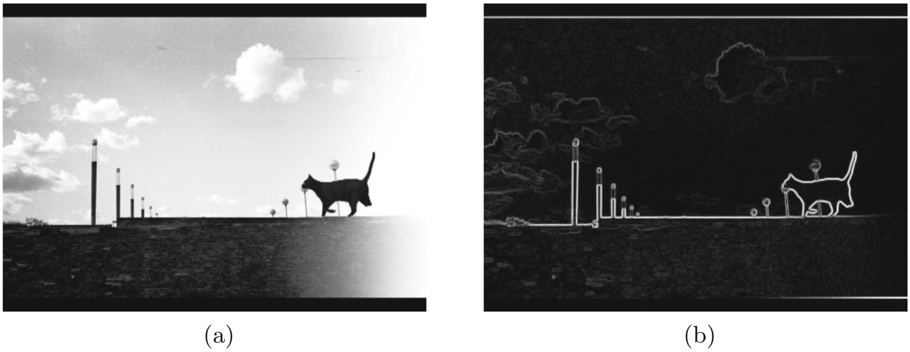


Fig. 3. (a) Frame 104 of the video (b) Sobel-filtered frame 104.

3.4 Building a Bounding Box Around Salient Objects

Having reached this point of the pipeline, the expected output are 2 CA configurations showing salient objects meant to be evaluated in the process of motion detection. In order to do this, a bounding box will be constructed around the ROIs and thus we will be able to collect their centroids and process an approximate estimation of the frame-to-frame behavior of the salient object.

The effectiveness of the estimation will be calculated upon completion of the collection of salient objects' centroids. A trajectory of all of the bounding boxes will show the approximate behavior of the moving object in the whole video.

4 Experimental Results

To exemplify what has been explained so far, the whole pipeline has been developed in pure Python language, using SciPy (ndimage)[1] library for the Sobel filtering part along with OpenCV[2] for several tasks on the video processing.

4.1 Analyzed Videos and Achieved Results

For evaluating the effectiveness of the approach, we used a video[3] with no camera movement, whose frame resolution is 360×496 pixels; the background is therefore permanently motionless (unless for artifacts due to video compression, changes in the illumination, etc.). The video represents a cat entering the screen from the right side and moving towards the other end. It must be noted that we did not run benchmarking tests for the analysis of computational times yet: in this work we mainly focus on the effectiveness of the approach, and its potential regarding the parallelization aspect will be considered in future works.

[1] https://docs.scipy.org/doc/scipy/reference/ndimage.html.
[2] https://opencv.org/.
[3] https://www.youtube.com/watch?v=HDb9StNG8_Q.

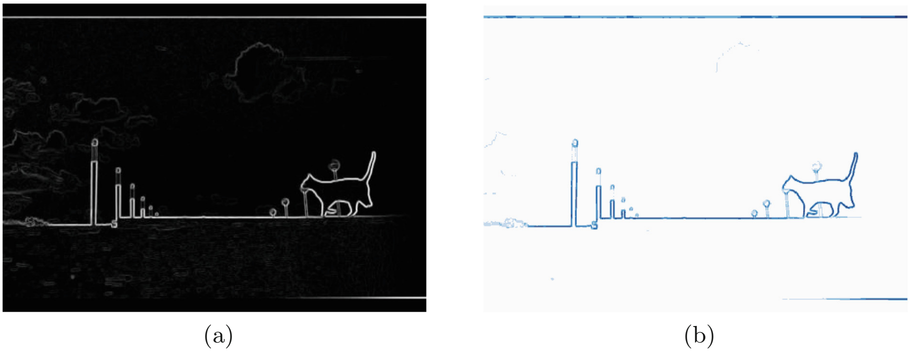


Fig. 4. (a) Sobel-filtered frame 104 (b) Flattened Sobel-filtered frame given as a CA configuration

From a Frame to a Sobel-Filtered Frame

In Fig. 3 it is shown how the Sobel operator works: given an image as input, it returns the most significant edges of that image based on their magnitude in terms of contrast.

CA Initialization

In Fig. 4 the Sobel-filtered frames was flattened to be better processed in the subsequent step of frames comparison. This step aims to remove superfluous edges, not so worth further evaluation.

Frames Comparison

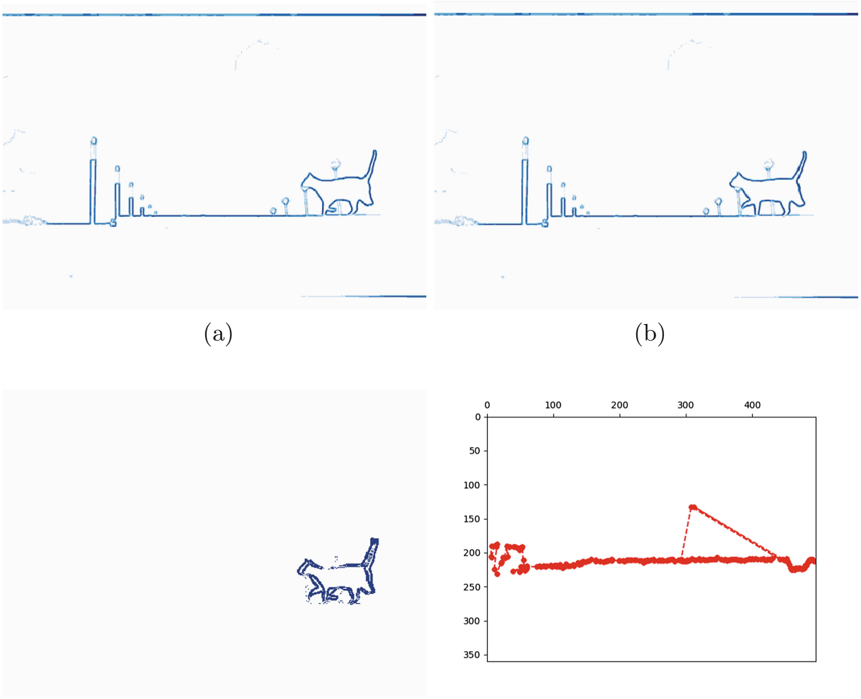
In order to better evaluate the difference between frames, we propose, in Fig. 5, 2 examples of differences through overlapping frames

Bounding box of Regions of Interest and their trajectories

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 *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 correctly does not report any movement. In Fig. 5d we show the positions of centroids of the bounding boxes built around ROIs.

In Fig. 6 we more briefly describe the results of another experiment, in which a video of a ball bouncing on screen[4], from the left side to the right side, was analyzed. Figure 6b shows the trajectory of centroids of ROIs of the video with a ball bouncing along the frame.

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


(a)

(b)

(c)

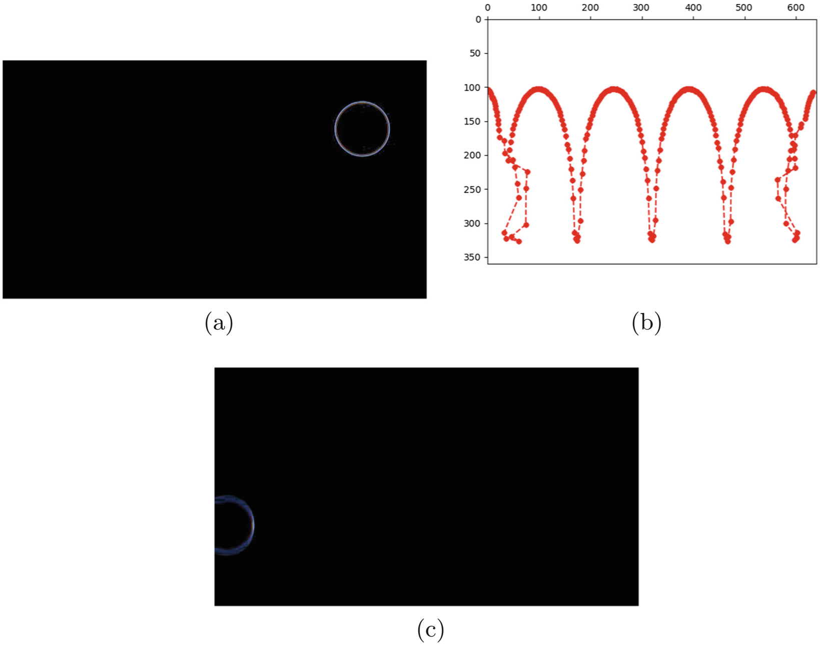
(d)

Fig. 5. (a) CA configuration of frame 104 (b) CA configuration of frame 106 (c) $\Lambda(L(F^{104}), L(F^{106}))$ (d) Trajectory of centroids of ROIs (markers identify centroids of bounding boxes of ROIs)

4.2 Discussion of Experiments

The heterogeneous movement of the cat and its tail provide a continuous although smooth change in the produced bounding box around the ROI, and this makes it quite dynamic and unstable. While the movement of the cat was basically homogeneous and predictable, the movement of its tail instead was fairly unpredictable. This lead to a continuous change of bounding boxes shapes. It is a matter of fact that this pointed out different movement directions between the cat and its own tail.

Moreover, the video presents some issues in terms of compression artifacts, leading to a slight change of colors of pixels in certain frames. On top of that, this method does not consider the problem of object classification, meaning that it does not consider the case of more objects moving in the same frame yet. Nevertheless, as it can be seen in Fig. 5, only 3 frames out of 219 show a clear discrepancy between the expected bounding box position and the one retrieved from the system: the points around the coordinates (300, 150) are due to the



(a)

(b)

(c)

Fig. 6. (a) A frame taken from a video of a ball bouncing on screen (upper right part of the frame) (b) Trajectory of ROIs of video with a bouncing ball (markers identify centroids of bounding boxes of ROIs) (c) 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).

recognition of noisy pixels in the top left part of the video as a possible moving object and part of the ROI.

The second test is proposed on another video that represents a ball with a black background bouncing at a static bouncing rate and moving from left to right at a constant speed. In this case, the object is fundamentally not changing from a morphological perspective, although it is constantly changing velocity, even with relatively significant displacements withing the frame. Results for this scenario are slightly more satisfactory than the previous experiment: even though 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 (see the points at the borders of Fig. 6b). This is due to the fact that, in those frames, e.g. Fig. 6a, the ball speed is rather high and its edges become blurry. This makes the Sobel filter face some difficulties in processing the gradient of ball edges. Therefore only the right edge of the ball is detected and the bounding box built around it makes the centroid of the bounding box slightly shifted along the two axes.

With reference to the achieved results in both the experiments, even before moving in the direction of trying to classify the detected objects, simply considering some physical constraints characterizing the typically observed objects (or the movement capabilities of an autonomous robot on which the camera is positioned) supports the possibility of completely dismissing or significantly reducing this kind of error. For instance, in [11] the authors analyzed trajectories generated by pedestrians and they were able to reject as outliers tracks in which changes of direction were simply too sudden for a walking human, but analogous considerations could be done with respect to commonsense reasoning [4] on the morphology of the detected and tracked objects.

5 Future Works

The present paper fundamentally reports the current results of an ongoing work investigating a wider research challenge, that is, 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, in the vein of [1,13].

The present results show that CA can represent useful blocks within a more complex work-flow for the processing of videos, in particular with the aim of detecting and characterizing motion within the analyzed frame. Relationship between the present model and current biological results are still thin; nonetheless, there are results related to the functioning of individual photo-receptors [6] and the conjecture is that CA could be applied to explain the visual processing on the retina. Visual processing is basically composed of local interaction between nearby photo-receptor cells at receptor level and inter-neurons at higher levels.

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 computationwise.

An additional work that could be taken as inspiration for future implementations is also [9], describing 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.: Using insects to drive mobile robots–hybrid robots bridge the gap between biological and artificial systems. Arthropod Struct. Dev. **46**(5), 723–735 (2017)
2. Avidan, S.: Support vector tracking. IEEE Trans. Pattern Anal. Mach. Intell. **26**(8), 1064–1072 (2004)
3. Canny, J.: A computational approach to edge detection. In: Readings in Computer Vision, pp. 184–203. Elsevier (1987)
4. Davis, E., Marcus, G.: Commonsense reasoning and commonsense knowledge in artificial intelligence. Commun. ACM **58**(9), 92–103 (2015)

5. Deriche, R.: Optimal edge detection using recursive filtering. Int. J. Comput. Vis. **2**, 167–187 (1987)
6. Frye, M.: Elementary motion detectors. Curr. Biol. **25**(6), R215–R217 (2015)
7. Georgoudas, I., Kyriakos, P., Sirakoulis, G., Andreadis, I.: An FPGA implemented cellular automaton crowd evacuation model inspired by the electrostatic-induced potential fields. Microprocess. Microsyst. **34**(7), 285–300 (2010)
8. Guo, J., Ren, T., Huang, L., Liu, X., Cheng, M.M., Wu, G.: Video salient object detection via cross-frame cellular automata. In: 2017 IEEE International Conference on Multimedia and Expo (ICME), pp. 325–330. IEEE (2017)
9. Hartbauer, M.: Simplified bionic solutions: a simple bio-inspired vehicle collision detection system. Bioinspiration Biomim. **12**(2), 026007 (2017)
10. Ioannidis, K., Andreadis, I., Sirakoulis, G.C.: An edge preserving image resizing method based on cellular automata. In: Sirakoulis, G.C., Bandini, S. (eds.) ACRI 2012. LNCS, vol. 7495, pp. 375–384. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33350-7_39
11. Khan, S.D., Bandini, S., Basalamah, S.M., Vizzari, G.: Analyzing crowd behavior in naturalistic conditions: identifying sources and sinks and characterizing main flows. Neurocomputing **177**, 543–563 (2016)
12. Kumar, T., Sahoo, G.: A novel method of edge detection using cellular automata. Int. J. Comput. Appl. **9**(4), 38–44 (2010)
13. Linan-Cembrano, G., Carranza, L., Rind, C., Zarandy, A., Soininen, M., Rodriguez-Vazquez, A.: Insect-vision inspired collision warning vision processor for automobiles. IEEE Circ. Syst. Mag. **8**(2), 6–24 (2008)
14. Popovici, A., Popovici, D.: Cellular automata in image processing. In: Fifteenth International Symposium on Mathematical Theory of Networks and Systems, vol. 1, pp. 1–6 (2002)
15. Prewitt, J.M.: Object enhancement and extraction. Pict. Process. Psychopictorics **10**(1), 15–19 (1970)
16. Qin, Y., Lu, H., Xu, Y., Wang, H.: Saliency detection via cellular automata. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 110–119. IEEE (2015)
17. Roberts, L.G.: Machine perception of three-dimensional solids. Ph.D. thesis, Massachusetts Institute of Technology (1963)
18. Rundo, L., et al.: Neuro-radiosurgery treatments: MRI brain tumor seeded image segmentation based on a cellular automata model. In: El Yacoubi, S., Wąs, J., Bandini, S. (eds.) ACRI 2016. LNCS, vol. 9863, pp. 323–333. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44365-2_32
19. Sobel, I.: An isotropic 3 × 3 image gradient operator. In: Machine Vision for Three-Dimensional Scenes, pp. 376–379 (1990)
20. Toffoli, T., Margolus, N.: Cellular Automata Machines: A New Environment for Modeling. MIT Press, Cambridge (1987)
21. Wolfram, S.: Cellular automata as models of complexity. Nature **311**(5985), 419–424 (1984)
22. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. (CSUR) **38**(4), 13 (2006)