A Computer Vision System for Monitoring Ice-Cream Freezers

Alessandro Torcinovich[1(✉)], Marco Fratton[2], Marcello Pelillo[1,3], Alberto Pravato[2], and Alessandro Roncato[1,2]

[1] DAIS, Ca' Foscari University, Via Torino 155, 30172 Venezia Mestre, Italy
840284@stud.unive.it
[2] PROSA S.r.l., via dell'Elettricità 3/d, 30175 Venezia-Marghera, Italy
[3] ECLT, Ca' Foscari University, S. Marco 2940, 30124 Venice, Italy

Abstract. In this paper, we describe a computer vision system aimed at monitoring the evolution of the content of a commercial ice-cream freezer. In particular, the system is able to detect the volume occupied by ice-creams in a basket and to track ice-cream sales. To this end, three modules have been developed performing the detection of the baskets and the products inside them, along with the tracking of the interactions with the freezer to take/drop products. The system comprises four cameras connected to an embedded mini-computer able to communicate with a telemetry system that sends information about the freezer context. Our proposed methods achieve promising results for the basket detection and the product tracking (accuracy around 70–80%) and good results in the volume estimation.

Keywords: Edge detection · Image segmentation · Optical flow estimation · Ice-cream freezers

1 Introduction

The need of a remote automated monitoring system for vending machines is mainly motivated by three facts. On a small scale, a more frequent analysis of vending machines content can help logistic activities plans, avoiding superfluous check-and-restock travels. On a larger scale, the collection of sales data coming from several vending machines allows more accurate sales strategies, regarding, for instance, what kind of products are sold in a particular geographical zone. Further, minor problems such as thief control and unwanted products detection, can be addressed as well.

To the best of our knowledge, there are not publicly documented projects with similar aims in the vending machines field, thus in this work we propose a novel system to address them. The effort has been focused on developing three modules devoted to perform two kinds of estimation, regarding: (a) the volume occupied by the ice-creams currently inside a basket (b) the amount of ice-creams sold in a particular time frame.

S. Battiato et al. (Eds.): ICIAP 2017, Part II, LNCS 10485, pp. 333–342, 2017.
https://doi.org/10.1007/978-3-319-68548-9_31

The former estimation has been addressed through the application of an interactive image segmentation approach, in particular a scribble-based one, to segment the *ice-cream zone*. To this end, Price et al.'s *geodesic graph cut* method [7] revealed to be the proper choice. However, since the system needs to work without the user aid, a further step was introduced to automatically draw the scribbles, based on the detection of the basket edges within which the ice-cream zone is.

As for the latter estimation, it has been developed a simple system able to detect the interaction between hands and ice-cream baskets, in particular the gestures related to a product being taken or being put into the basket. This has been accomplished through a motion tracker, based on a background subtractor and an optical flow estimator.

The data collection was performed by four fish-eye cameras, each placed above a basket. Fish-eye was required in order to capture as much visual information as possible.

The system have been tested with many configurations trying to emulate real use cases, and performed its required tasks with a good accuracy and efficiency.

2 Description of the Modules

In this section the algorithmic details of the developed modules are presented. At the beginning, the system checks for basket edges, then it performs periodical checks of the basket product level. At the same time, the freezer is constantly monitored to check for potential interactions and when they are found, they are asynchronously analyzed to detect gestures. The reader can refer to Fig. 1 for a sketch of the system pipeline outlining the interaction between the modules to perform the aforementioned tasks.

2.1 Basket Edges Detection (BED) Module

This module searches for peculiar parts of the baskets in the image, from now on referred as *basket edges*. In particular, the basket edges which we are interested in are the three visible *top borders* defining the rim of the basket, and the *bottom*. The basket is assumed to be in *overflowing state* if the top borders are not visible, and in *empty state* if the bottom is visible. It is important to note that the baskets are not fixed and the empty space between two of them can be up to 5 cm.

As for the top borders, our approach is inspired by the road-lines detection method explained in [1], since the two vertical top borders bear a lot of resemblance to road-lines.

First, the image is color-thresholded in order to filter out all non-white intensities. To this aim the image is converted to HSV color-space and then only the pixels with high V-values and low S-values are retained, more precisely all the pixels in the range $(\mu_l - \alpha_l\sigma_l, \mu_l + \beta_l\sigma_l)$, where μ_l and σ_l are the mean and standard error of the l-values in the image ($l \in \{S, V\}$), while $\alpha, \beta > 0$ are chosen

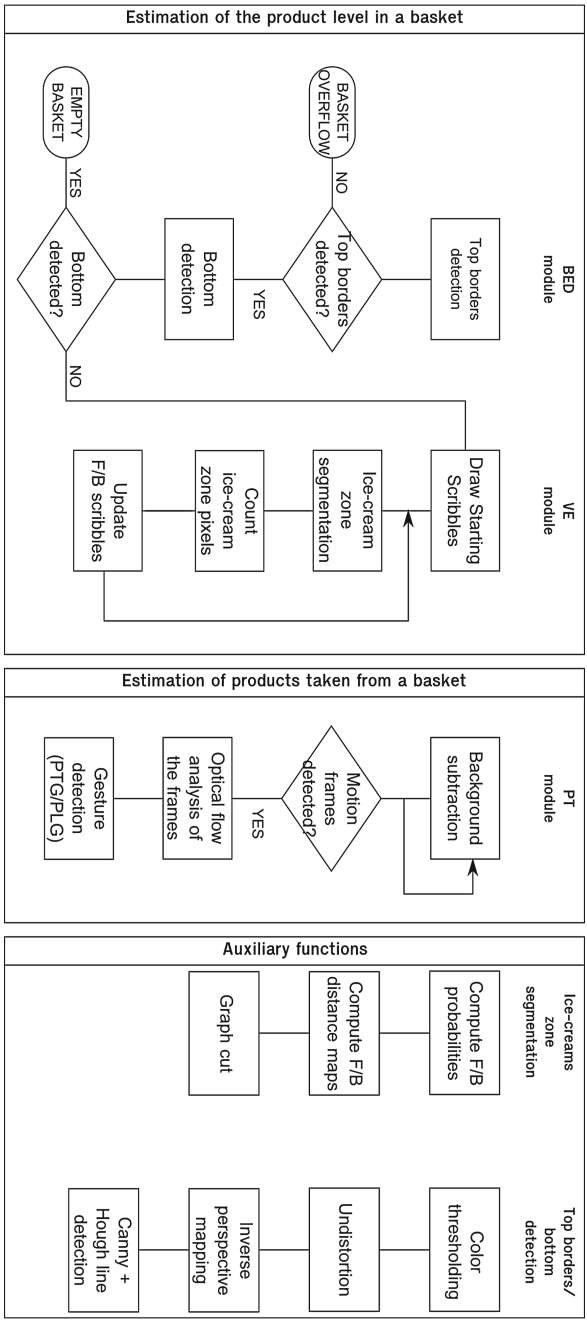

Fig. 1. A simplified sketch of the system pipeline

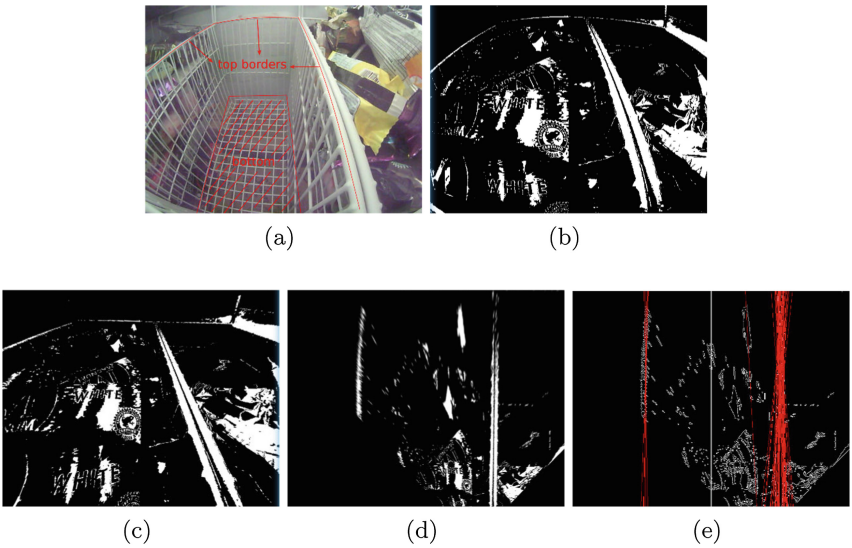

Fig. 2. (a) The basket edges to be detected. (b) The color mask after the color thresholding phase. (c) The undistorted image. (d) The bird's eye view. (e) Applying Hough transform on the Canny filtered version. (Color figure online)

heuristically. Sometimes occasional light occlusions can selectively darken one zone of the basket, possibly causing the vertical top borders to have very different associated V-values. To overcome this problem the thresholding is applied separately on the left and the right half of the image.

After this, the image is undistorted and then its perspective is canceled through an *Inverse Perspective Mapping (IPM)* procedure [1,3]. Further, a Canny filter [5] is applied to obtain the edges in the resulting image.

The following step consists in the detection of vertical lines in the image, that is accomplished through the *Hough Transform line detection* algorithm [9]. Among all the possible candidates, only a pair of lines has to be selected. The choice is made taking into consideration the following restrictions:

- The top borders under current investigation are vertical lines (constraint on θ).
- One of them is in the left half of the image, the other in the right half. They cannot be in the center of the image (constraint on ρ).
- The distance between the top borders must be consistent with the effective size of the basket (26 cm ca).

These constraints improve the search of a good pair of lines matching the top vertical borders.

After finding the two lines, their upper ends are joined to draw the third top border, then perspective and distortion are applied to. Figure 2 shows the main steps of the algorithm and the detected lines for a typical case.

The bottom is detected in a similar way, focusing only on the area bounded by the top borders, but horizontal lines—instead of vertical ones—are searched. In particular the algorithm gives separates responses for the upper and lower part of the bottom, depending on the number of detected horizontal lines in the two areas.

2.2 Volume Estimation (VE) Module

The VE module searches for the ice-cream zone within the basket edges found in the previous steps. Its implementation relies on the segmentation of the image through the approach described in [7], which basically proposes a revised version of the graph cut algorithm [4] with a cost function based on the geodesic segmentation [2]. The cooperation between the two algorithms results in a more accurate segmentation, aware of both the spatial and the colour intensity information. In summary, the so-called *Geodesic Graph Cut* works minimizing the following cost function:

$$E(\mathcal{L}) = \sum_{x_i \in P} R_{\mathcal{L}_i}(x_i) + \lambda \sum_{(x_i,x_j)\in N} B(x_i, x_j)|\mathcal{L}_i - \mathcal{L}_j| \tag{1}$$

where, $\mathcal{L} = (\mathcal{L}_i)$ is a binary vector of labels and $\mathcal{L}_i \in (\mathcal{F}, \mathcal{B})$ determines if a pixel is labeled as a foreground or background one. Here, $B(x_i, x_j)$ is a *boundary cost-term*, computed as:

$$B(x_i, x_j) = \frac{1}{1 + \|C(x_i) - C(x_j)\|^2} \tag{2}$$

$R_l(x_i)$ is instead a *region term*, computed as

$$R_l(x_i) = s_l(x_i) + M_l(x_i) + G_l(x_i) \tag{3}$$

where

$$s_l(x_i) = \begin{cases} \infty & \text{if } \mathcal{L}_i = l \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

while $M_l(x_i) = P_{\bar{l}}(C(x_i))$ represents the probability that the color intensity associated to the pixel x_i belongs to the opposite label of l (if $l = \mathcal{F}$, then $\bar{l} = \mathcal{B}$).

The last term represents a normalized geodesic distance, in particular:

$$G_l(x_i) = \frac{D_l(x_i)}{D_{\mathcal{F}}(x_i) + D_{\mathcal{B}}(x_i)} \tag{5}$$

Figure 3 shows the pixel maps of the second and third terms, for both the foreground and background.

In order to speed up the computations the paper suggests to use approximate methods such as the Fast Gauss Transform [11] for the second term and the

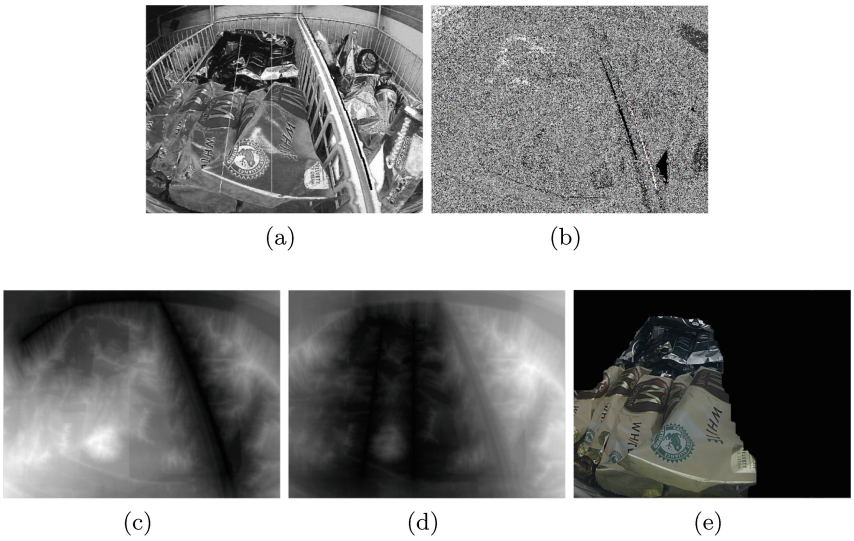

Fig. 3. (a–b) The probability pixel maps, foreground and background respectively, related to the second term in Eq. 3 (scribble colors are inverted to be more visible in the images). (c–d) The normalized distance maps related to the third term in the same equation. (e) An image segmented with the proposed method.

Fast Marching Method [2,8] for the third one. In addition, further accuracy improvements can be made (refer to [7] for more details).

The method requires the user to define an initial input representing a sample of background pixels and another one of foreground pixels. As previously noted, the output of the BED module is used to set the background strokes, while the foreground ones are drawn within the basket area. Once the first segmentation is performed, the ice-cream zone is dilated and the borders of the resulting area are taken as the new background strokes.

2.3 Product Tracking (PT) Module

The last module identifies significant interactions within the freezer and tries to guess what happened. The core of the module is based on two algorithms to detect movement. At first, a simple background subtractor [12] is run to capture movement. Whenever some activity is detected, the analysis is passed to optical flow detection algorithm until a period of overall inactivity is detected, where the background subtractor takes control again. The necessity to rely on two motion detection algorithms is due to the fact that the optical flow algorithm is resource-demanding, and cannot be run in real-time in the embedded system, however it is strictly needed because it is able to estimate the direction of the flow[1]. The

[1] Currently Farnebäck's method [6] is used, but in the future, to overcome this problem, it will be replaced with some faster technique like Tao's SimpleFlow [10].

frames between two inactivity periods are assumed to contain information about what kind of interaction has been performed within the freezer. Currently two gestures are detected, namely the *Product Taken* (*PTG*) and the *Product Left* (*PLG*) ones, denoting if an ice-cream has been taken from/put into the basket.

To detect a gesture, the pixels in a frame are first divided in four classes: (a–b) *up/down motion pixels* associated to motion vectors with a high norm and a direction pointing up/down, (c) *undefined motion pixels* associated to motion vectors with a low norm or a non vertical direction (or both), (d) *inactive pixels* where no motion is detected.

After classifying the pixels, the algorithm discriminates between *up/down motion frames*, simply checking what is the highest between up/down motion pixels. The next step consists in checking the highest amount of down pixels among all down frames against the respective amount among all up frames. If an ice-cream has been taken, it is expected that the max amount of up pixels, representing the hand and the ice-cream, is higher with respect to the max amount of down pixels representing the hand only, and the algorithm outputs a PTG response. The opposite can be stated for an ice-cream put into a basket.

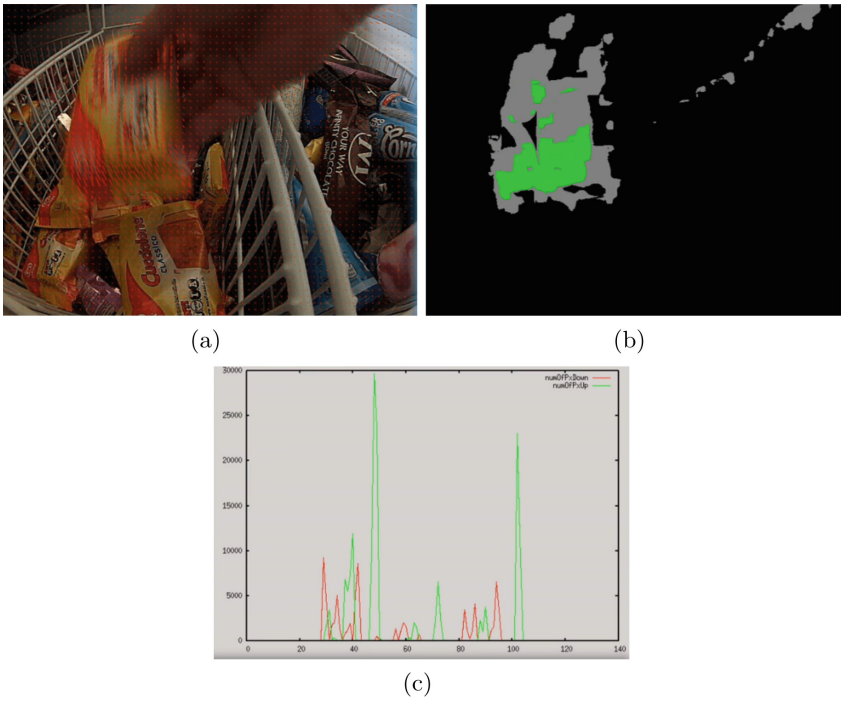

Fig. 4. (a) The optical flow of a video frame during an interaction with the freezer. (b) The corresponding pixel classification in up (green) and undefined (gray) motion pixels. Inactive pixels are colored in black, while down motion pixels are not present. (c) Plot representing the amount of down (red) and up (green) pixels in each frame of a video containing three PTGs. (Color figure online)

As an example, in Fig. 4(c) the amount of up/down pixels is shown for a sequence of three PTGs. Each gesture can be identified from the pattern of a down pixels peak, followed by an higher up pixels peak.

3 Experimental Results

A prototype of the ice-cream freezer was assembled, and we tested the three modules separately with multiple freezer contexts evolving over the time. All the tests were performed with several brands and different lighting conditions on a system with a dual core CPU with 3 GHz/core and a 2 GB RAM, however in the future further testing will be performed under more restrictive conditions. The running time of a volume analysis performed by the BED + VE modules stays under 12 s, while for the gesture detection, the time is equal to 130% of the video length containing the gesture.

The BED module testing was performed searching for the basket edges in 55 situations, comprising baskets with different product levels. A visual inspection of the results, revealed that the top borders detection was successful in $48/55 \approx 87.3\%$ of cases. In particular, sometimes the algorithm failed in the case of baskets with a top border next to the white inner wall of the freezer, because of the similarity in colour intensity. The bottom detection was performed only in case that the top borders have been found. Since the tests comprised an important percentage of overflow settings, the actual number of bottom detection tests decreased to 39 cases. The percentage of correct detections was $30/39 \approx 76\%$. Further testing revealed that also the bottom detection can suffer from similarity in colour intensity with the inner walls.

Some of the VE module tests are shown in Fig. 5. The initial scribbles were correctly drawn by the BED module and then the geodesic graph cut was

Fig. 5. Examples of VE analyses performed on an evolving context in the freezer.

iteratively applied. After tuning the number of dilations, the ice-cream zone was correctly detected with high precision without problems.

The PT module was tested analyzing 9 manually captured videos containing 66 PT/PL gestures. In the 73% of the cases the gesture was correctly detected and classified. The module was further tested with a setting in which a torch light was switched on and off periodically to simulate sudden lighting changes. The algorithm responded well, without erroneously detecting gestures.

4 Conclusions and Future Work

In this paper, we have described a suite of algorithms devoted to monitor the content of an ice-cream commercial freezer, to provide information about the content level and the products taken from the freezer, reaching good results for both the tasks. The project is still in a prototype stage and some of its aspects needs to be refined, in order to cover a wider majority of the real cases.

The BED module can be improved introducing the detection of basket dividers, which usually are freely movable and can be put to divide between multiple brands inside a basket. As for the PT module the algorithm as is, was tested using only one camera per basket, however being able to monitor the interactions in the fridge with two cameras could possibly increase the effectiveness of the algorithm and give us a more accurate position of the hand in the basket. This could be paired with a noise analyzer to detect the particular sound that a hand produces when scratching in search of ice-creams, providing us even more information about the interaction. The VE module, strongly depends on the drawn scribbles and assumes a progressive emptying of the basket. Indeed it can happen that the basket is subject to some "brutal" modification, like the re-fill or a total emptying, that can potentially invalidate the successive volume check. This can be dealt by pairing the module with the other two, especially with the product tracking module, making it capable to recognize these particular situations. The module currently work if the first check is performed with an adequately full basket. This aspect needs improvements in order that the first volume check can be run successfully with any possible basket status.

References

1. Aly, M.: Real time detection of lane markers in urban streets. In: 2008 IEEE Intelligent Vehicles Symposium, pp. 7–12. IEEE (2008)
2. Bai, X., Sapiro, G.: Geodesic matting: a framework for fast interactive image and video segmentation and matting (preprint). Technical report, DTIC Document (2008)
3. Bertozzi, M., Broggi, A.: Real-time lane and obstacle detection on the gold system. In: Proceedings of the 1996 IEEE Intelligent Vehicles Symposium, pp. 213–218. IEEE (1996)
4. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 1, pp. 105–112. IEEE (2001)

5. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 679–698 (1986)
6. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003). doi:10.1007/3-540-45103-X_50
7. Price, B.L., Morse, B., Cohen, S.: Geodesic graph cut for interactive image segmentation. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3161–3168. IEEE (2010)
8. Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. Proc. Natl. Acad. Sci. **93**(4), 1591–1595 (1996)
9. Shapiro, L., Stockman, G.: Computer Vision. Prentice Hall, Upper Saddle River (2001)
10. Tao, M., Bai, J., Kohli, P., Paris, S.: Simpleflow: a non-iterative, sublinear optical flow algorithm. Comput. Graph. Forum **31**, 345–353 (2012). Wiley Online Library
11. Yang, C., Duraiswami, R., Gumerov, N.A., Davis, L.S., et al.: Improved fast gauss transform and efficient kernel density estimation. In: ICCV, vol. 1, p. 464 (2003)
12. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 2, pp. 28–31. IEEE (2004)