Provisional chapter

Human Activity Recognition without Vision Tracking

Carlos Alberto Flores Vázquez, Joan Aranda,
Daniel Icaza, Santiago Pulla,
Marcelo Flores-Vázquez and
Nelson Federico Cordova

Additional information is available at the end of the chapter

Abstract

This work describes the recognition of human activity based on the interaction between people and objects in domestic settings, specifically in a kitchen. The difference between this and other proposals is that considers a human activity in a process without vision tracking. Videos are a sequence of photographs. Taking this into account, if you analyze an orderly sequence of images it could be based on the objects present in each scene so that you can understand the possible activity performed. However, it is not enough to consider the objects present in the scene; it is necessary to determine if those objects are employed or not by the humans present. If they are used, it is evident that they are necessary to carry out the activity; if they are not used they would only provide noise to the recognized activity. Therefore, it is necessary to generate a conceptualization of objects in the scene with characteristics (definition of an object, motion detector, object recognition, object position, object action) that allows you to recognize them and to determine the degree of use (unchanged, added, removed, moved, and indeterminate) and influence the possible recognized activity.

Keywords: human activity, computer vision, human/computer interaction, human/robot interaction, feature extraction, behavior representation

1. Introduction

This study is part of a major research project called InHands [1, 2]. The approach specifically analyzes the recognition of human activity without using the traditional proposal that uses the

follow-up of movements of hands and arms. A short version of this research is shown in [3] and this work presents several improvements to the initial proposal presented in [2].

To introduce the field of recognition of human activity is of interest in this research and the definition mentioned in [4] on human/object interactions is as follows:

"The most typical human-object interaction recognition approaches are the ones ignoring interplays between object recognition and motion estimation. In those works, objects are generally recognized first, and activities involving them are recognized by analyzing the objects' motion. They have made the object recognition and motion estimation independent or made it so that the motion estimation is strictly dependent on the object recognition" [2, 4].

In addition to the definition quoted here, this chapter defines the structure and classification of human activity recognition, from which we extract the following:

- Single-layered approaches are appropriate for gesture and action recognition by sequential characteristics [2, 4].
- Hierarchical approaches are human activity representations with a high level of abstraction. Within this topic we find a subclassification that is of interest to us [2, 4]:
    - Statistical: construct statistical state-based models concatenated hierarchically [2, 4].
    - Syntactic: use a grammar syntax such as stochastic context-free grammar to model sequential activities [2, 4].
    - Description-based: represent human activities by describing subevents of the activities and their temporal, spatial, and logical structures [2, 4].

This research applies the approach of human/object interactions, and included in these are more specific subtopics: syntactic and description-based.

In [5] a proposal for the recognition of activity based on description-based is presented. This methodology consists of motion detection and tracking complemented by event analysis, which is of interest for the detection of movement. On the basis of this the capture of images for our proposal is carried out. As in [5] we use a single camera for capture and for segmentation the background is extracted, but we use image difference.

Among the relevant works to establish an activity recognition procedure [6], each action event is assigned a symbol and then a sequence of actions corresponds to a string of symbols. In our proposal we will use words instead of symbols, and a set of words regardless of their order could form an activity.

A similar way to [6] is the proposal of [7]: a BOW (bag of words). A BOW considers that an image might be similar to a paragraph where repetition of one or more words would allow us to recognize the content or essence of the text. For us it is similar to considering the repetition of objects in the image, and to interact with them would give us indications of the activity that is developing.

For the InHands project, proactive assistance is very important, and with this premise in [8] they present a probabilistic prediction of the actions carried out, which is precisely what is intended to implement our proposal.

A similar work where the scenario is to employ a Kinect camera is [9]. However, to increase the reliability of this methodology of recognition radio-frequency identification tags are added, which will not be implemented in our proposal.

The first information this system considers is hand and object tracking, and later object and action recognition. Regarding the detail of the recognized actions, seven principals are defined: place; move; chop; mixing; pouring; spooning; and scooping [9].

An example of the results obtained is the preparation of a cake, for which seven objects, 17 actions, about 6000 frames, and approximately 200 seconds are used. For us the definition of actions is simpler. Therefore, our actions will be those explained in Section 2.1.5.

2. Methodology

In this research, we consider that the results of activity recognition would be useful to provide proactive assistance. Therefore, the recognition of the activity should be determined while the activity is being carried out and with the aim of facilitating the robotic assistance considered in later stages of the InHands project.

Recognition is approached with computer vision methods. In a specific way, it is based on the recognition of objects in the scene and the interaction with these objects based on their manipulation. However, it is necessary to detail that the interaction with the objects does not contemplate the tracking of the objects, arms, or hands of the person who intervenes in the action. The initial and final positions of the objects, their presence or not, are of importance in the recognition process.

2.1. Conceptualization of an object

In this proposal the conceptualization of an object is explained in five constitutive parts:

- Definition of an object
- Motion detector
- Object recognition
- Object position
- Object action

2.1.1. Definition of an object

Considering the relevance of the objects and interaction with these, it is necessary to develop a definition of the object with parameters that allow the differentiation between them. Therefore, the chosen parameters are:

1. *Identification number*: this allows us to have a unique number for each object despite having similar characteristics such as color.

2. *Color*: this is the main characteristic that allows us to recognize the type of object present in the scene.
3. *Position*: this is confirmed by the coordinates of the centroid for each object, taking as origin the lower left corner of the kitchen counter.
4. *Actions with objects*: four basic interactions have been defined with the objects by the user: add, remove, move, unchanged.

Figure 1(a) graphically shows the conception of an object in this proposal. **Figure 1(b)** shows the execution of the algorithm in parallel to generate the attributes of each object.

2.1.2. Motion detector

Motion detection is important so that we know that an activity is taking place in the scene. In addition, objects in the scene (OBJECT RECOGNITION) and their position (OBJECT POSITION) are acquired. Since we do not track, is important to recapture the objects in the scene after the movements made by the person present. By making this subsequent capture we avoid occlusions and we can determine an action (OBJECT ACTION) for each object when checking through their presence and position if they were moved, removed, added, or not used in the last action.

As demonstrated in the flow chart of **Figure 2**, one of the most important methods applied in this part of the system is image difference, specifically "the mixture of the Gaussian method" according to [10]. **Figure 2(a)** explains the flow chart for motion detection and **Figure 2(b)** shows the three results after executing the motion detection algorithm.

This image difference allows us to extract the objects from the background, which is dynamically updated while the system is working. The motion detection algorithm performs a continuous comparison of frames by setting a minimum threshold level to consider whether that variation between a frame at $t = 0$ and the following at $t + 1$, $t + 2$, and $t + 3$ implies an action performed or is only a visual noise.

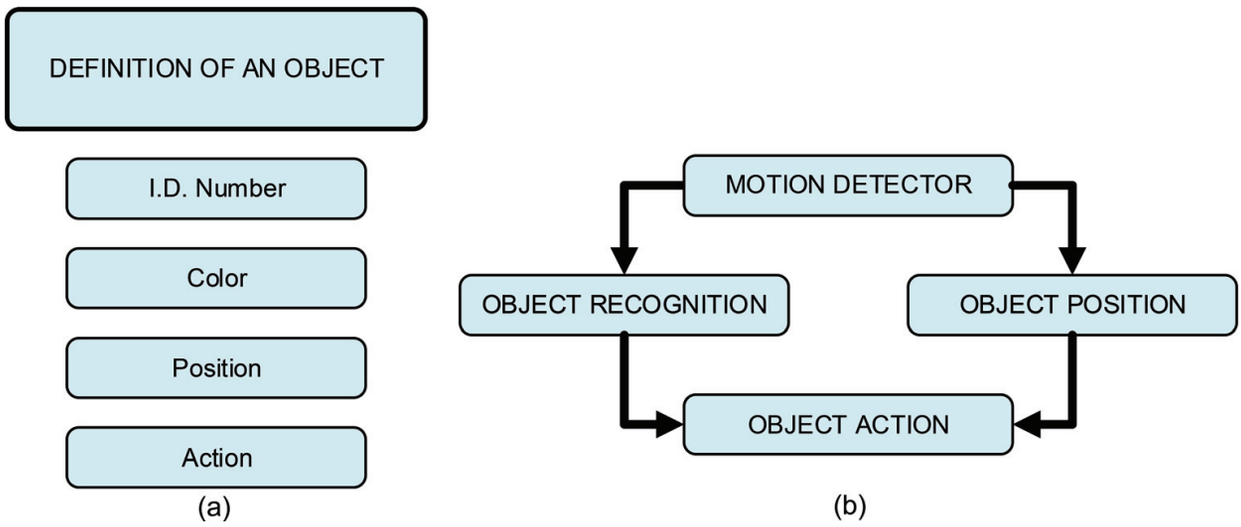

Figure 1. (a) Graphical model of our definition of an object. (b) General flow chart of the definition of an object.

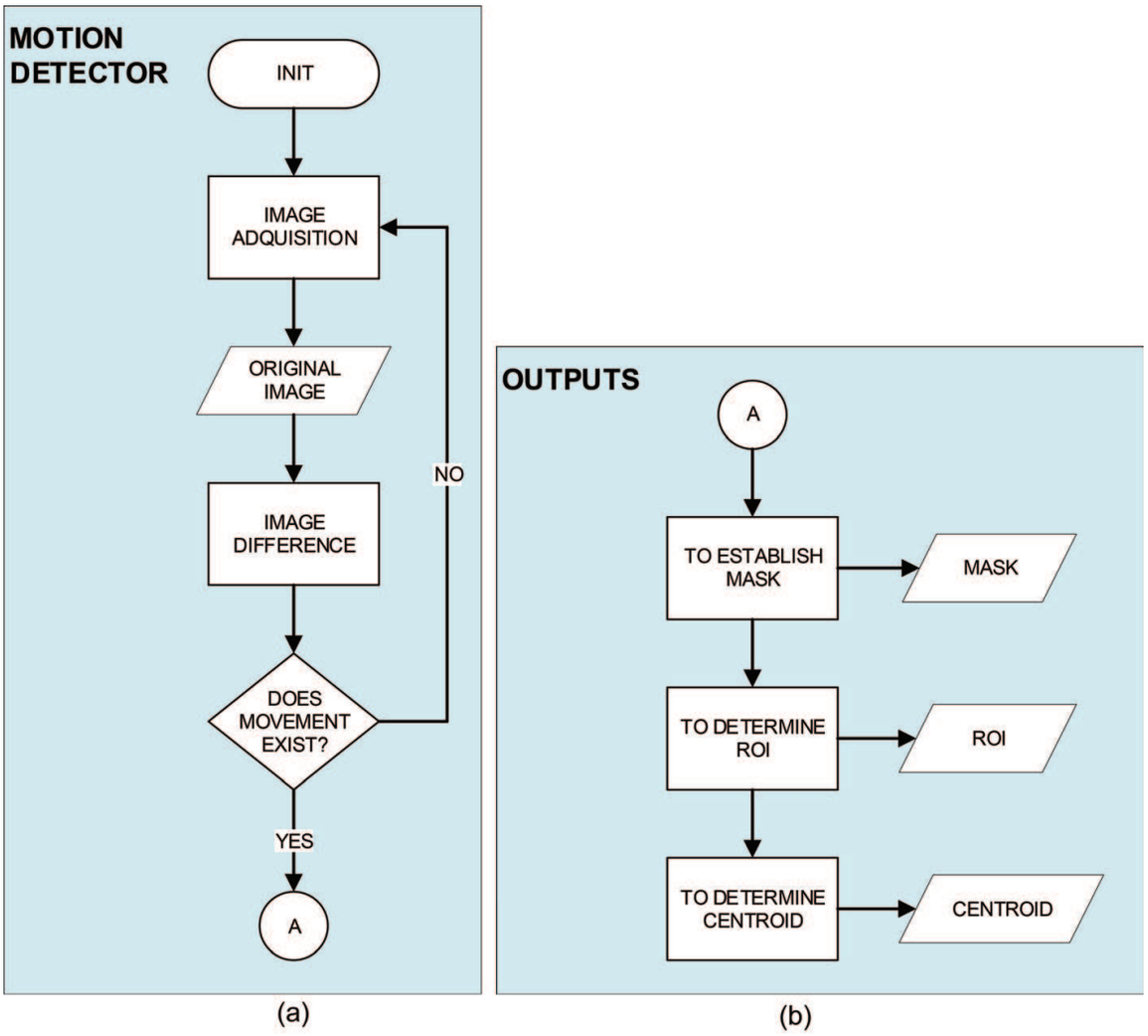

Figure 2. Flow charts: (a) motion detector, (b) outputs from motion detector.

2.1.3. Object recognition

The object recognition process is developed in **Figure 3**. The first step is to get the mask by difference of images and apply it to the original image. The result of the application of the mask is the region of interest (ROI).

From each ROI we obtain a histogram of 10 BINS RG chromaticity space (two-dimensional color space in which there is no color intensity information). Working with RG chromaticity allows us to avoid the problems of brightness, additionally normalized the images in RGB color space (a pixel is identified by the intensity of red, green and blue values) and was restricted by thresholds the colors black and white.

A comparison of the new histograms obtained from the ROI was made against our database. The selected comparison method was the Bhattacharyya distance. As evidenced in **Table 1** this method was selected considering the percentage of errors reduced and low amount of time.

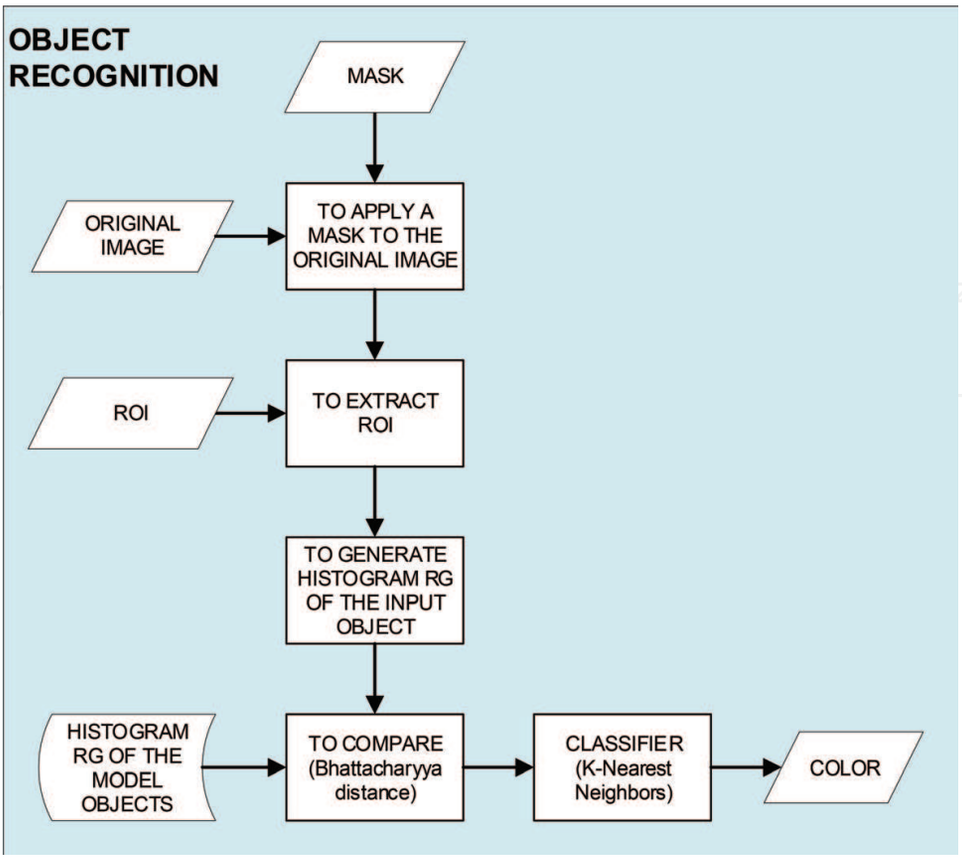

Figure 3. Flow chart for object recognition.

Method	Correlation	Chi-square	Intersection	Bhattacharyya distance
Samples	100	100	100	100
Error	8%	8%	2%	2%
Observations [5]	Quick	Moderately fast—more accurate matches	Quick—and—dirty matching	Moderately fast—more accurate matches
Range [exact … mismatch]	[1.0 …–1.0]	[0.0…2.0]	[1.0…0.0]	[0.0…1.0]

Table 1. Comparison of methods.

The database of the objects has five different perspectives for each one. As far as the classifier was concerned, K_{nn} (nearest neighbors) was chosen with $K = 1$.

Evaluation of the aforementioned classifier was performed in a similar way to the proposal in [11]. In detail, a confusion matrix is employed to measure the recognition of each of the objects. Forty images were used for validation and 30 for each test of a total of 1870 images. Some examples are shown in **Table 2**.

Probably a better alternative to reduce time in object recognition is [12]. The YOLO Detection System is a really fast detector that can process streaming video in real time with less than 25 ms

Coffee		**Predicted label**		**Measure**	**Result**
		Positive	Negative	Precision	91.67%
Known label	Positive	22	4	Recall/sensitivity	84.62%
	Negative	2	2	Specificity	50.00%
				Accuracy	84.62%
Glass		**Predicted label**		**Measure**	**Result**
		Positive	Negative	Precision	100.00%
Known label	Positive	24	5	Recall/sensitivity	82.76%
	Negative	0	1	Specificity	100.00%
				Accuracy	82.76%
Spoon		**Predicted label**		**Measure**	**Result**
		Positive	Negative	Precision	100.00%
Known label	Positive	23	2	Recall/sensitivity	92.00%
	Negative	0	5	Specificity	100.00%
				Accuracy	92.00%

Table 2. Confusion matrix and measurements: (a) coffee, (b) glass, (c) spoon.

of latency. However, the YOLO method is not convenient for localization errors, for example Fast R-CNN has 8.6% of localization errors versus 19% for the YOLO method. This is explained in Section 2.1.4; localization is an important part of this proposal for object definition.

Other alternatives taking to account reduced time can be [13, 14]. Tensor flow is a flexible system and can be used to express a wide variety of algorithms. For this proposal, the main advantage would be the capacity for distributing the process in many computational devices for object recognition.

2.1.4. Object position

In this proposal the position of the object is one of the essential characteristics; this allows us to define the actions resulting from the interaction with it. If the position does not change between images it means that the object was not used (UNCHANGED); if on the other hand it changes it means that the object is necessary in the developed activity (MOVE).

As for the technical details the centroid is a pixel in the image; this pixel is positioned in the coordinate system of the image. Taking into account that the InHands project requires obtaining world coordinates to assist with robots, then these coordinates must be referenced to the kitchen counter.

Homography matrix H is used according to [15]. It is necessary to apply matrix H; the intrinsic and extrinsic matrixes of the calibration of the camera are explained in [16]. The result of homography is expressed in millimeters and the final adjustment is made with rotation and translation matrixes (**Figure 4**).

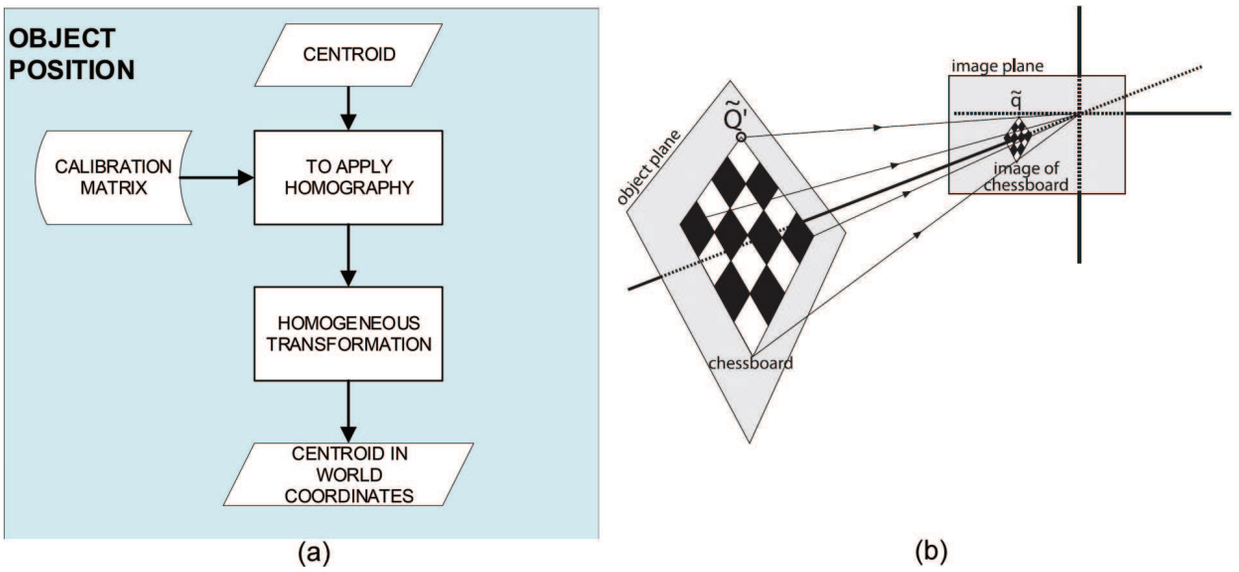

Figure 4. (a) Flow chart of object position. (b) View of a planar object as described by homography [15].

2.1.5. Object action

To build the object with the previously obtained characteristics (ID number, color, centroid), in "Action" we assign the state of "UNDETERMINED," see **Figure 5(a)**. The human/object interaction is defined in the feature of the object called Action. This can take four possible options:

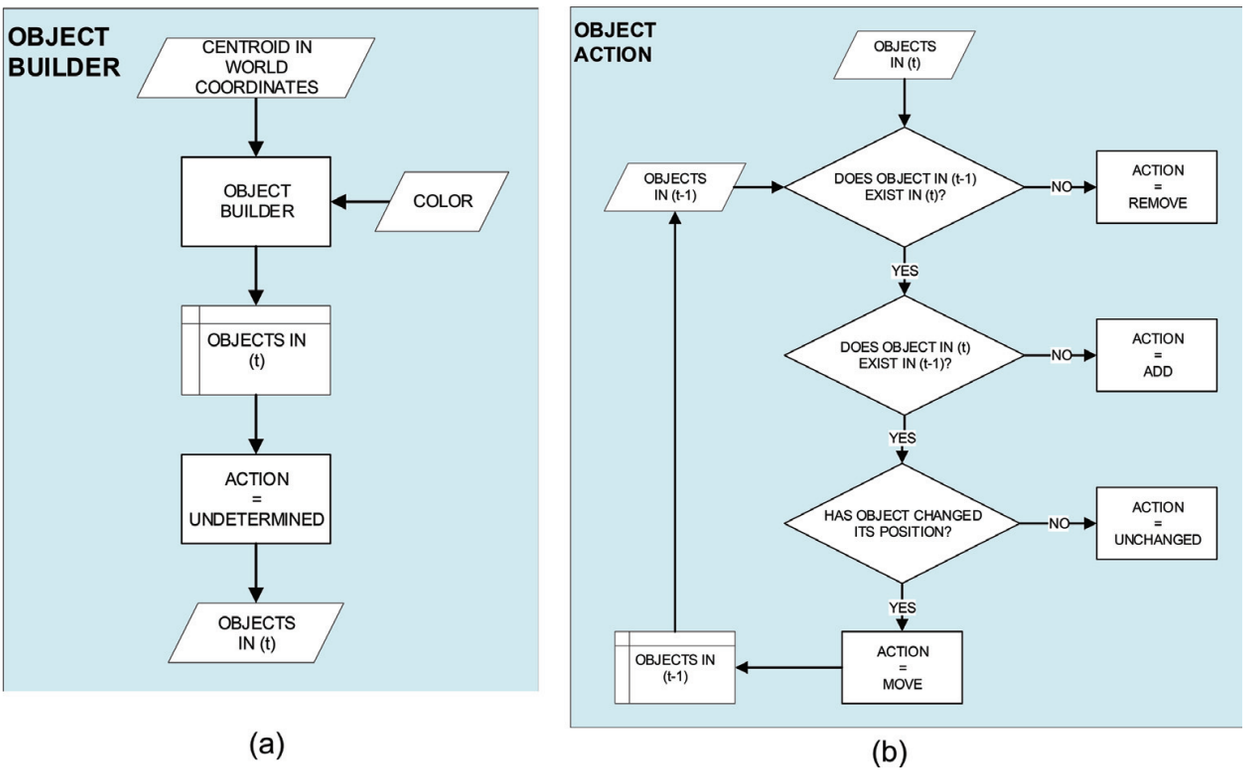

Figure 5. (a) Flow charts of object builder. (b) Flow charts of object action.

1. UNCHANGED: The object was present in the previous activity but it has not been moved, which implies that the object is present in the scene but it was not used in the activity.
2. MOVE: This indicates that the object was present in a previous scene and now changes position implying that it was used in the developing activity.
3. ADD: The object was not present in previous scene, and because it is now added to the scene it is assumed to be necessary for the developing activity.
4. REMOVE: An object present in previous scenes is no longer present. This induces the thought that it is now not necessary for the activity that is developing.

To avoid detecting false movements caused by occlusions or errors in the calculation of the centroid a tolerance range was established; only movements greater than 5 mm are recorded.

2.2. Human activity recognition

The taxonomy proposed in [4] allows us to illustrate an approximation to the approach that this research has. In detailed form it is typecast in hierarchical approaches and has many coincidences with the vision presented in syntactic and description-based. Specifically, the approach of human/object interactions uses a syntax to define human activity but it is not necessary to consider order, sequences, and logical structure (**Figure 6**).

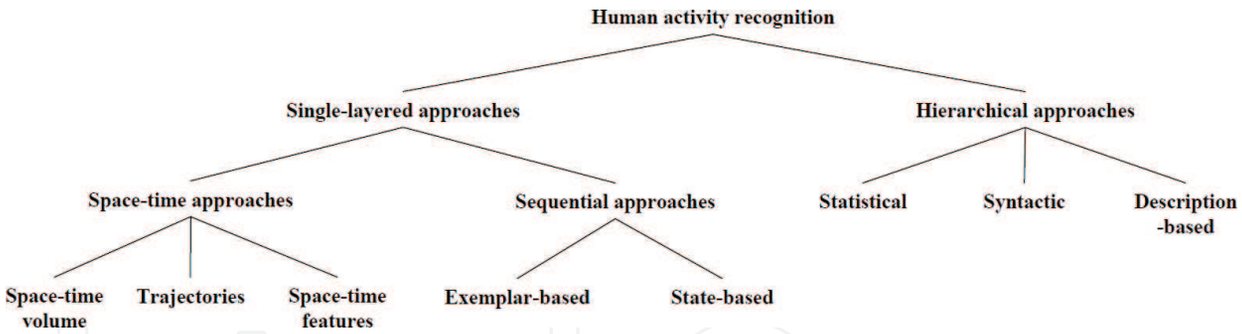

Figure 6. The hierarchical approach-based taxonomy [4].

An approach whereby it is important to make a comparison of the methodology used is [7]. BOW collects features by assigning the nearest word and the frequency of occurrence of this in the images. In our proposal, each object could be a word and the repetition of these words will be relevant to determine the activity, but unlike [7] it is not necessary to consider the sequence of occurrence of words.

2.2.1. Definition of an activity

How do you define an activity? This is one of the questions that arose during the project; in this case a recipe inspires it, so we will use ingredients, kitchen tools, and possible substitutes to define an activity, see **Figure 7(a)**.

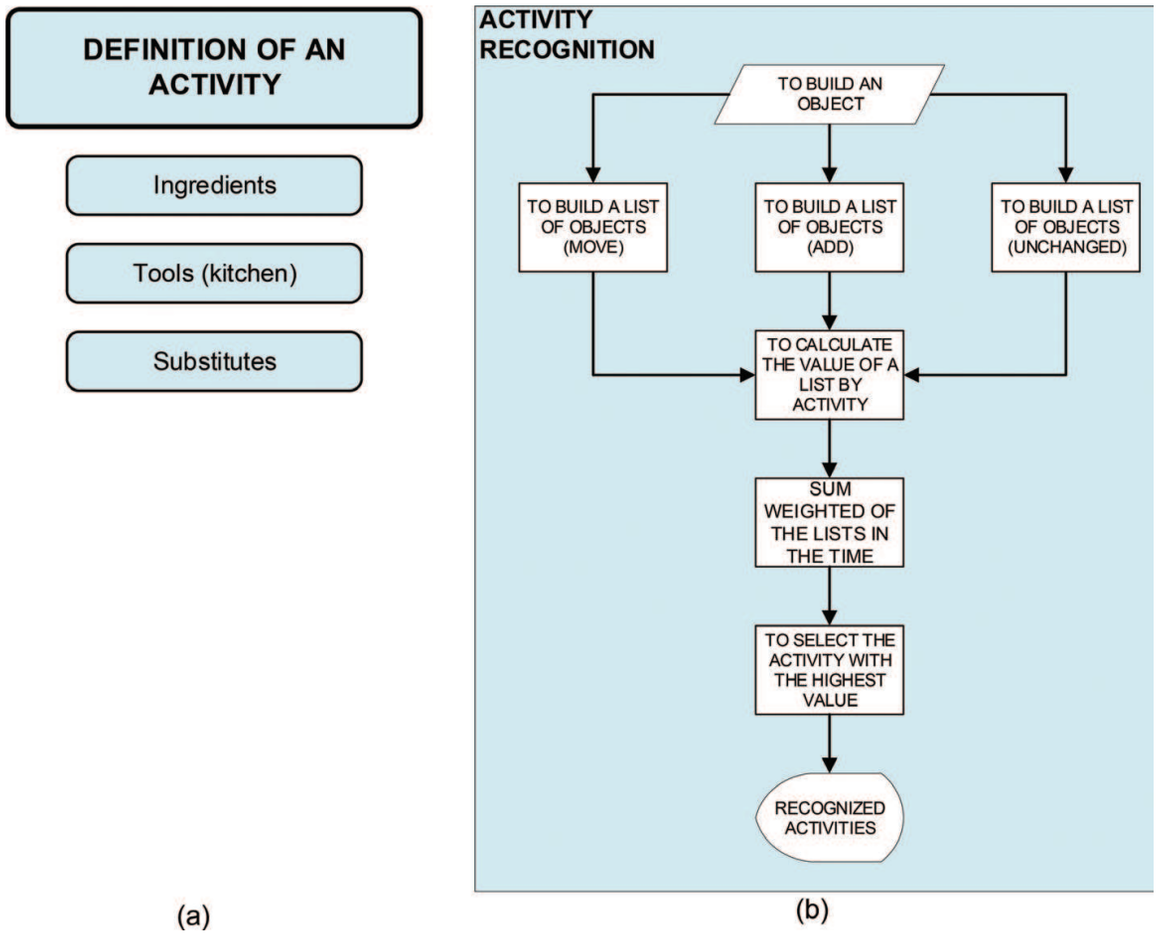

Figure 7. (a) Definition of an activity. (b) Flow chart of activity recognition.

- INGREDIENTS: This refers to objects considered as ingredients for the preparation of a recipe, e.g. for a cereal-activity (cereal, milk).
- TOOLS: This includes cooking utensils and cutlery necessary for the elaboration of the recipes-activities, e.g. cereal-activity (bowl, spoon).
- SUBSTITUTES: These are kitchen utensils that could be replaced by others that perform a similar function, e.g. a cup for a glass.

2.2.2. Evaluation function and activity recognition

To start the recognition of activity we made three groupings of objects according to their action, in other words a list for moved objects (MOVE), another one for objects added to the scene (ADD), and a list of objects present in the scene, but they do not have to be moved or withdrawn (UNCHANGED).

The first evaluation corresponds to calculating that the contribution of each ingredient, utensil, and substitute has undergone a movement, taking into account that the contribution of ingredients, utensils, and substitutes is being weighted by the constants a, b, and c, respectively. The result is the value that these objects provide for each of the probable activities carried out Eq. (1).

The second evaluation corresponds to calculating the contribution of each ingredient, utensil, and substitute that has been added to the scene. Similar to the first evaluation, the contribution of ingredients, utensils, and substitutes is considered to be weighted by the constants a, b, and c, respectively. The result is the value that these objects provides for each of the probable activities carried out Eq. (2).

The third evaluation is exactly the same in its procedure as the two previously made, with the only difference being that the objects that intervene here are the ones that have remained in the scene without any change, i.e. they have not been moved, added, or removed (Eq. 3).

It is important to emphasize that an object could be a utensil or a substitute depending on the activity, for example a glass would be a utensil if the activity is to prepare juice (activity 1) but would be a substitute if the activity is to prepare coffee (activity N). where:

$$\begin{bmatrix} V_{Act\_1} \\ \vdots \\ V_{Act\_N} \end{bmatrix}_{[M]} = a \cdot \begin{bmatrix} I_{Act\_1} \\ \vdots \\ I_{Act\_N} \end{bmatrix}_{[M]} + b \cdot \begin{bmatrix} T_{Act\_1} \\ \vdots \\ T_{Act\_N} \end{bmatrix}_{[M]} + c \cdot \begin{bmatrix} S_{Act\_1} \\ \vdots \\ S_{Act\_N} \end{bmatrix}_{[M]} \tag{1}$$

$$\begin{bmatrix} V_{Act\_1} \\ \vdots \\ V_{Act\_N} \end{bmatrix}_{[A]} = a \cdot \begin{bmatrix} I_{Act\_1} \\ \vdots \\ I_{Act\_N} \end{bmatrix}_{[A]} + b \cdot \begin{bmatrix} T_{Act\_1} \\ \vdots \\ T_{Act\_N} \end{bmatrix}_{[A]} + c \cdot \begin{bmatrix} S_{Act\_1} \\ \vdots \\ S_{Act\_N} \end{bmatrix}_{[A]} \tag{2}$$

$$\begin{bmatrix} V_{Act\_1} \\ \vdots \\ V_{Act\_N} \end{bmatrix}_{[Un]} = a \cdot \begin{bmatrix} I_{Act\_1} \\ \vdots \\ I_{Act\_N} \end{bmatrix}_{[Un]} + b \cdot \begin{bmatrix} T_{Act\_1} \\ \vdots \\ T_{Act\_N} \end{bmatrix}_{[Un]} + c \cdot \begin{bmatrix} S_{Act\_1} \\ \vdots \\ S_{Act\_N} \end{bmatrix}_{[Un]} \tag{3}$$

- V_{Act} = Result for each activity from 1 to N.
- I_{Act} = Recognized objects that are considered ingredients for each activity.
- T_{Act} = Recognized objects that are considered utensils for each activity.
- S_{Act} = Recognized objects that are considered substitutes for each activity.
- $[M, A, Un]$ = MOVE, ADD, UNCHANGED.
- a, b, c = Constants for tuning contribution, $a = 0.5$, $b = 0.3$, $c = 0.2$ [2].

The fourth evaluation corresponds to the addition of the activity lists resulting from Eqs. (1)–(3). Explicitly the result of Eq. (4) (SUM WEIGHTED OF THE LISTS IN THE TIME) corresponds to adding the probable activities that result from moving, adding, or not changing objects in a scene (Eq. 4). This result would correspond to the probable instantaneous activity, i.e. in the last frames (1 to 4 frames).

$$\begin{bmatrix} \sum V_{Act\_1} \\ \vdots \\ \sum V_{Act\_N} \end{bmatrix} = \alpha \cdot \begin{bmatrix} V_{Act\_1} \\ \vdots \\ V_{Act\_N} \end{bmatrix}_{[M]} + \beta \cdot \begin{bmatrix} V_{Act\_1} \\ \vdots \\ V_{Act\_N} \end{bmatrix}_{[A]} + \gamma \cdot \begin{bmatrix} V_{Act\_1} \\ \vdots \\ V_{Act\_N} \end{bmatrix}_{[Un]} \tag{4}$$

where:

- $\sum V_{Act}$ = The summation of value by activity (the last 1 to 4 frames).
- α, β, γ = Variables changing on the time, $\alpha + \beta + \gamma \leq 1$.

$$\alpha = \frac{1}{3} + \left(\frac{1}{6} - \gamma\right) \tag{5}$$

$$\beta = \frac{1}{3} + \left(\frac{1}{6} - \gamma\right) \tag{6}$$

$$\gamma = \frac{1}{3} \bullet \left(\frac{ElapsedTime}{AverageTime}\right) \tag{7}$$

where:

- $ElapsedTime$ = The elapsed time from the start of an activity. Initial value is set to $AverageTime$, later decreased.
- $AverageTime$ = Average time for the execution of any predefined activity.

Factors α, β, γ serve to ponder the contribution of $[V_{Act}]_M$, $[V_{Act}]_A$, $[V_{Act}]_{Un}$. Note that activities resulting from actions such as moving (MOVE) or adding (ADD) objects to the scene are more relevant over time than the result of just keeping objects in the scene with no position changes (UNCHANGED).

The results of Eq. (4) represent activities recognized in the last four frames, so in no way would represent a global or final result of the activity recognized.

To obtain a more reliable result we must add a considerable group of results of Eq. (4), and by adding these results we obtain a statistically reliable result of the activity recognized activity, Eq. (8) being the maximum value of the activity recognized.

$$Activity\_Recognized = \max\left\{\sum_{1}^{Tsamples}\begin{bmatrix}\sum Act\_1\\ \vdots \\ \sum Act\_N\end{bmatrix}\right\} \tag{8}$$

Tsamples = Total samples of results of Eq. (4) during average time of activity recognized.

3. Results

The complete system that is obtained to perform our proposal of activity recognition is outlined in **Figure 8**; each of its constitutive parts was explained in the previous sections.

For experiments in a real domestic scenario, we count on the automated kitchen developed under the InHands project in **Figure 9(a)** [1].

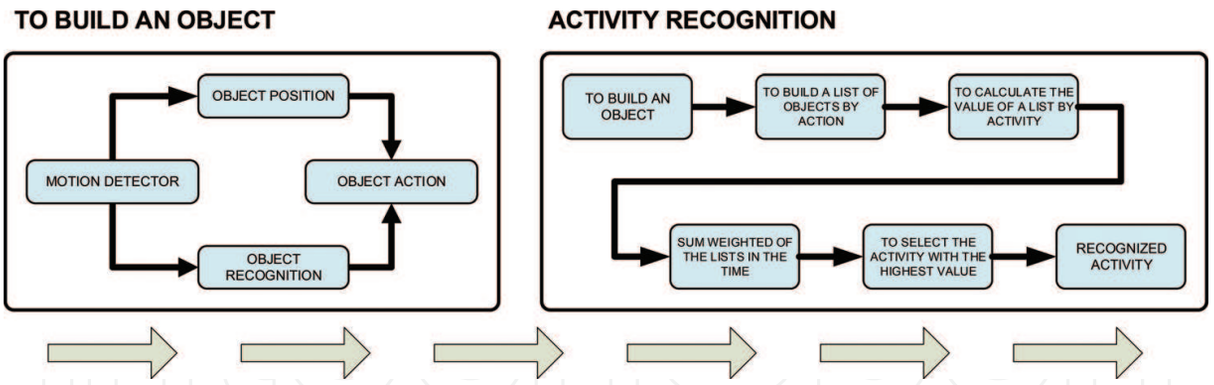

Figure 8. Complete implemented system.

Figure 9. (a) InHands automated kitchen scenario. (b) Selected objects for the experiment.

In this research a color Kinect camera was employed; the depth of view was not used. People doing the cooking activities in **Figure 9(a)** did so randomly without any training or prepared script. The system was tested with four typical breakfast activities consisting of: brewing coffee, juice preparation, cereal preparation, and chocolate preparation.

The ingredients, cooking utensils, and substitutes are shown in **Figure 9(b)** as follows: coffee, sugar, chocolate, juice, cereal, milk, bowl, cup, glass, plate, and spoon.

The first tests of the system of recognition of the proposed activity used five videos for each of the four activities raised. The results were excellent and after 600 frames it could be clearly differentiated which was the activity being executed in **Figure 10(b)**. Partial results of Eq. (4) are illustrated in **Figure 10(a)**. It should be mentioned that the results of 1 to 4 frames illustrated in **Figure 10(a)** suffer from occlusions and changes in lighting that hinder a more

accurate recognition of objects; however, thanks to the evaluation of Eq. (8) the erroneous partial results can be filtered to obtain a correct overall result.

Proactive support is one of the objectives of the InHands project, so our system should be able to recognize activities without having to be segmented, i.e. in a normal sequence of events recognize the different activities that are being developed.

As mentioned, the following test phase consisted of placing several activities without segmenting and checking whether the system was capable of recognizing them. **Figure 11** illustrates one of the tests performed with the preparation of unsegmented juice, cereal, and coffee without a preconfigured specific order. As evidenced in **Figure 11(a)** the instantaneous activity recognition system (1 to 4 frames) is unclear in showing the activity developed but after processing it with Eq. (8) the result is clear (**Figure 11(b)**). Therefore, the functioning of the systems in a continuous way without the need to segment activities is demonstrated. The

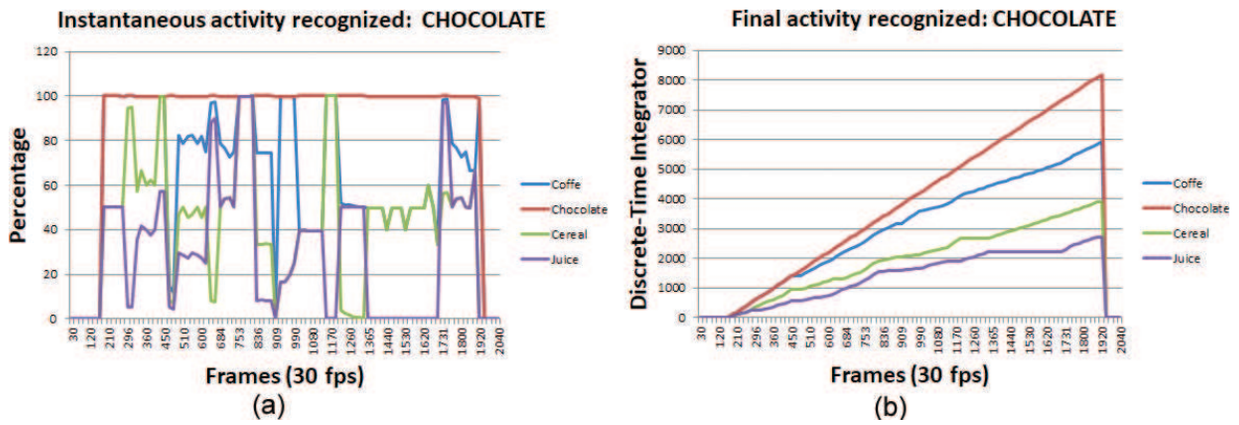

Figure 10. (a) Instantaneous activity recognized $(\sum V_{Act})$: CHOCOLATE. Vertical axis = percentage of similarity with the $(\sum V_{Act})$, horizontal axis = number of frames. (b) Final activity recognized $\left(\sum_{1}^{Tsamples}\right)$: CHOCOLATE.

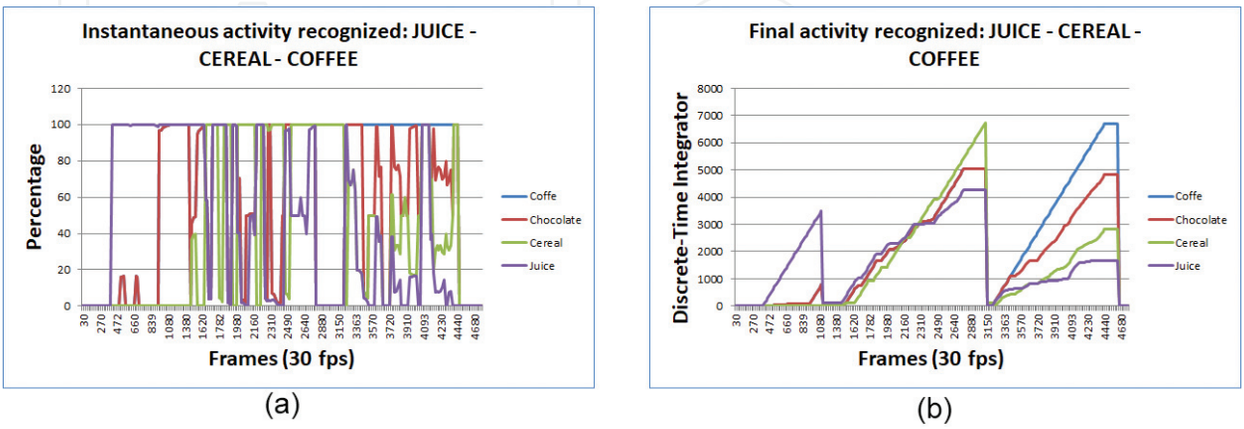

Figure 11. (a) Instantaneous activity recognized $(\sum V_{Act})$: JUICE—CEREAL—COFFEE. (b) Final activity recognized $\left(\sum_{1}^{Tsamples}\right)$: JUICE—CEREAL—COFFEE.

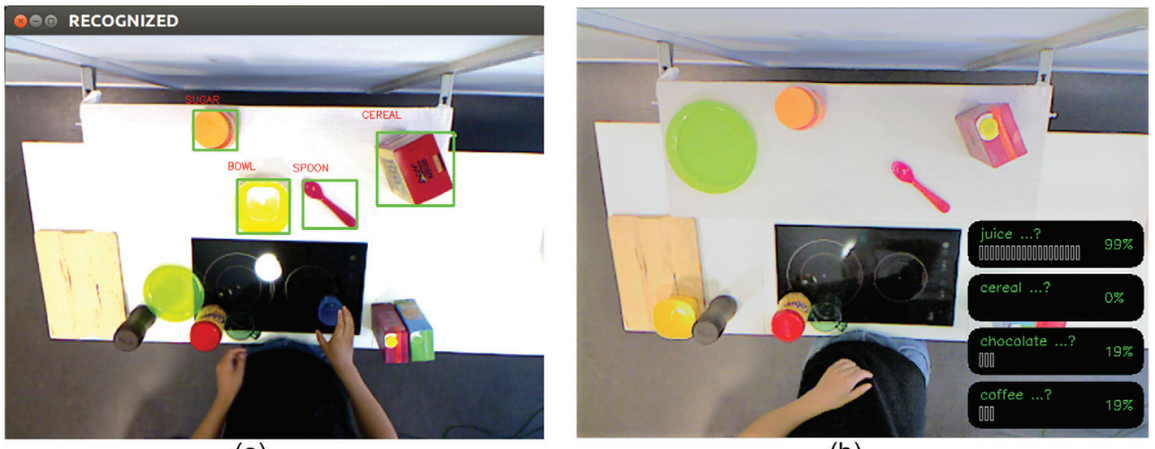

Figure 12. (a) Sample frame of a processed video sequence. (b) Sample frame of a processed video computer interface.

final activity is coffee preparation with satisfactory performance from the beginning. **Figure 12** shows a sample frame of our video process for the recognition of objects and activities [3].

4. Conclusions

This proposal for recognition of human activity is different from others based on tracking as shown in [17]. It is based on the preparation of cooking recipes, considering the interaction of the user with the objects present in the scene. In a detailed way, after the recognition of objects we classify them in categories (ingredients, utensils, and substitutes) and for the interaction we define four actions (add, remove, move, and unchanged) weighting the contribution of objects and interactions to determine possible activity.

Among its notable features are: it does not use intrusive methods with the user and requires an average time of 0.25 s for an instant recognition of activity [18–22]. It is also noted that in all the tests the recognitions were met by surpassing problems of brightness and occlusions to allow completely natural movements of the user.

The system is flexible and scalable by simply adding more activity definitions (recipes). The system works continuously with no default activity segmentation.

Future work would be to define with statistical methods the weighting constants here designated as a, b, c,α, β, γ.

Acknowledgements

This research was supported by the InHands project (Interactive robotics for Human Assistance in Domestic Scenarios), grant P6-L13-AL.INHAND founded by Fundaci La Caixa, inside the Recercaixa research program [3].

C. Flores and J. Aranda are associated with the Institute for Bioengineering of Catalunya and Universitat Politècnica de Catalunya, Barcelona-Tech, Spain [3].

Author details

Carlos Alberto Flores Vázquez[1]*, Joan Aranda[2], Daniel Icaza[1], Santiago Pulla[1], Marcelo Flores-Vázquez[3] and Nelson Federico Cordova[1]

*Address all correspondence to: cfloresv@ucacue.edu.ec

1 GIRVyP Research Group, Catholic University of Cuenca, Cuenca, Ecuador

2 IBEC Institute for Bioengineering of Catalonia, Polytechnic University of Catalonia, Barcelona–Tech, Barcelona, Spain

3 Salesian Polytechnic University, Cuenca, Ecuador

References

[1] Vinagre M, Aranda J, Casals A. An interactive robotic system for human assistance in domestic environments. In: International Conference on Computers for Handicapped Persons. Cham: Springer; 2014. pp. 152-155

[2] Flores Vázquez C. Human activity recognition from object interaction in domestic scenarios [MS thesis]. Universitat Politècnica de Catalunya; 2014. Available from: https://upcommons.upc.edu/bitstream/handle/2099.1/23706/TFM%20Carlos%20Flores%20Vazquez%2013062014.pdf?sequence=1&isAllowed=y [Accessed: 10-04-2018]

[3] Flores-Vázquez C, Aranda J. Human activity recognition from object interaction in domestic scenarios. In: Ecuador Technical Chapters Meeting (ETCM); IEEE. IEEE; 2016. pp. 1-6. Available from: https://upcommons.upc.edu/bitstream/handle/2117/100423/v2%2bHUMAN%2bACTIVITY%2bRECOGNITION.pdf?sequence=3&isAllowed=y [Accessed: 10-04-2018]

[4] Aggarwal J, Ryoo M. Human activity analysis: A review. ACM Computing Surveys. 2011; **43**(3):16

[5] Hongeng S, Nevatia R, Bremond F. Video-based event recognition: Activity representation and probabilistic recognition methods. Computer Vision and Image Understanding. 2004;**96**(2):129-162

[6] Moore D, Essa I. Recognizing multitasked activities from video using stochastic context-free grammar. In: Proceedings of 18th National Conference on Artificial Intelligence. 2002. pp. 770-776

[7] Liefeng B, Sminchisescu C. Efficient match kernel between sets of features for visual recognition. In: Advances in Neural Information Processing Systems (NIPS). 2009. pp. 135-143

[8] Ryoo M. Human activity prediction: Early recognition of ongoing activities from streaming videos. In: IEEE International Conference on Computer Vision (ICCV); IEEE; 2011. pp. 1036-1043

[9] Lei J, Ren X, Fox D. Fine-grained kitchen activity recognition using rgb-d. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing. Pittsburgh, Pennsylvania: ACM; 2012. pp. 208-211. http://dx.doi.org/10.1145/2370216.2370248

[10] Laganière R. OpenCV Computer Vision Application Programming Cookbook. 2nd ed. Birmingham, UK: Packt Publishing Ltd; 2014. pp. 272-277

[11] Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. In: Information Processing & Management. 2009. vol. 45, no. 4, pp. 427-437. https://doi.org/10.1016/j.ipm.2009.03.002

[12] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. pp. 779-788

[13] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Kudlur M. Tensorflow: A system for large-scale machine learning. In: OSDI. Vol. 16. 2016. pp. 265-283

[14] Szegedy C, Reed S, Erhan D, Anguelov D, Ioffe S. Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441. 2014

[15] Bradski G, Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. California, USA: O'Reilly Media, Inc.; 2008. pp. 201-204, 384-404

[16] Mordvintsev A, Abid K. OpenCV-Python Tutorials: Camera Calibration and 3D Reconstruction [Internet]. Available from: http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html#calibration [Accessed: 10-04-2018]

[17] Stauffer C, Grimson W. Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition; IEEE; 1999. pp. 246-252

[18] Goebel R. ROS by Example. Vol. 2. Lulu.com; 2015

[19] Team OpenCV Dev. OpenCV 2.4.13.6 Documentation [Internet]. Available from: http://docs.opencv.org/2.4/modules/refman.html [Accessed: 10-04-2018]

[20] Creative Commons Attribution, Ros Tutorials [Internet]. Available from: http://wiki.ros.org/ROS/Tutorials [Accessed: 10-04-2018]

[21] Python Software Foundation. The Python Tutorial [Internet]. Available from: https://www.python.org/ [Accessed: 10-04-2018]

[22] Dennis D, Tin C, Marou R. Color Image Segmentation [Internet]. Available from: https://theiszm.wordpress.com/tag/color-segmentation/ [Accessed: 10-04-2018]

IntechOpen