Human detection and tracking using a Kinect camera for an autonomous service robot

Luis Ferreira, Antonio Neves, Artur Pereira,
Eurico Pedrosa, and Joao Cunha

Transverse Activity on Intelligent Robotics,
IEETA/DETI, Universidade de Aveiro
Aveiro, Portugal
{lff,an,artur,efp,joao.cunha}@ua.pt
efp,joao.cunha}@ua.pt
http://www.ieeta.pt/atri/

Abstract. This paper presents a novel method for people detection and tracking using depth images provided by Kinetic camera. The depth image captured by a Kinect camera is analysed using its histogram, allowing for the depth image to be divided in slices, making the retrieval of regions of interest a simple and computationally light process when compared to point clouds. These regions are then classified as human or not using a template matching technique. An efficient gradient descent algorithm is used to perform the template matching, using the RPROP algorithm, and the tracking is performed based on color image histogram comparison for each region of interest, in consecutive frames. The proposed method is viable for on-line detection and tracking of people and has been tested in a mobile platform in an unconstrained environment.

Keywords: people detection, people tracking, depth image, template matching, kinect

1 Introduction

In order for a robot to interact with its surroundings it has to be able to do some basic tasks, being one of the most important to see and understands what its seeing. In recent years, many advances have been made in the Computer Vision research area, where some projects have been deployed and proven to be effective in the interaction between robots and humans.

The goal of the work presented in this paper is to create a usable solution for people detection and tracking, in an unconstrained environment used in a mobile platform, making future work focused in the interaction between robots and humans a possibility.

The Robot Operating System (ROS) middleware, and C++ in conjunction with the OpenCV framework were chosen to implement this project, while

the physical mobile platform is the CAMBADA's team service robot, CAMBADA@Home, built specifically for the @Home challenge present in Robocup competitions and used for academic research.

This paper is organized as follows: section 2 presents relevant works used as study cases for this project, section 3 presents an overview of the algorithm and it's operation, and section 4 draws some final remarks on the proposed system.

2 Related work

For the past ten years it can be seen an increase of activity in the social robotics research area. The improvement of mobility and processing power of todays computers allow for projects like domestic service robots to become more available as time goes by.

If we go back to some of the former works developed on people detection ([1], [2]) we can see a tendency to use techniques based on background extraction. These can be applied to any type of images (e.g. color, thermal or depth) and present good results on human object detection as long as they fulfil strict requirements (e.g. stationary camera or a model of the background).

The appearing of RGB-D cameras, such as the Microsoft Kinect or Asus Xtion, benefited many projects of the computer vision research area due to the availability of two types of images, color and depth, on the same device while maintaining a very low price when compared to other 3D or thermal cameras, such as Laser Range Finders or Long-Wave Infrared (LWIR) cameras. Some related works such as [3], [4] and [5], use this new type of cameras to perform human detection, identification and tracking.

There are two main approaches preferred by researchers when dealing with human detection. The first is based on machine learning methods, such as AdaBoost [6], [7], [8] or Support Vector Machines, that use features like Histogram of Oriented Gradients (HOG) [9] or Local Surface Normals (LSN) [10] to perform a classification of objects as human or non-human. Other widely used technique is template matching, employed by systems such as [1] or [4], and present solutions using different types of images.

3 Proposed algorithm

The algorithm presented in this paper makes use of novel methods, some inspired by existing work in the people detection research area, where the most influential would be Xia, L., et al. [4] work. A Kinect camera is used to capture the environment in the form of images, both from the depth camera and the color camera. The choice of using images instead of point clouds was mainly related to the computational cost, and the ability to use known image-processing techniques.

The image from the depth camera enables and facilitates the detection of shapes. The process starts by analysing the depth image's histogram in order to

detect relevant areas of the image characterized by peeks in the histogram. This enables the slicing of the scene retrieving only part of it in the form of 2D images, that are later analysed in order to determine possible regions of interest (ROIs). These ROIs are then classified as human or not using the RPROP algorithm [11], inspired by it's use as a localization method in [12], and now used as part of a template matching technique.

Tracking of ROIs classified as human is performed by histogram comparison on the color image. This technique has been proven to be fast and effective on relating the same ROIs across consecutive frames, even if the regions disappears and reappears due to detection errors, enabling tracking of multiple persons.

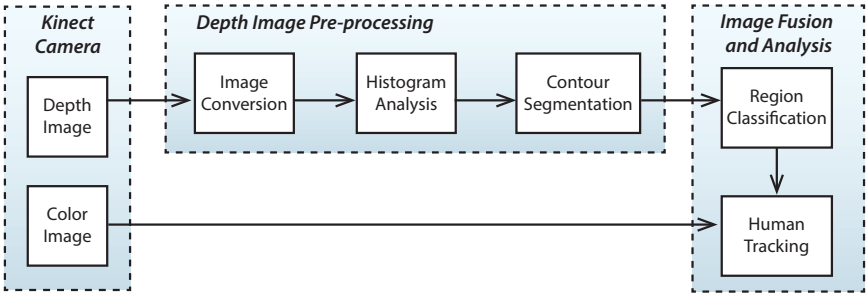


Fig. 1. Overview of the proposed method

3.1 Obtaining Regions of Interest

The first stage, and essential part of the algorithm, is the detection of regions of interest in the scene, also described here as ROIs. These work as masks indicating the regions of the image, both color and depth, that are populated by a possible human object.

Depth Image conversion The first stage, even before any image analysis, is necessarily the image acquisition. When working with the Kinect camera, this is an important part of the process, because throughout the proposed method 8-bit images are used, meaning 256 possible values for each pixel. However, the ROS driver for Kinect used to perform the communication between the camera and the program is only capable of delivering 16-bit images, 65536 different values per pixel, where each pixel carries the value for the distance of that point relative to the Kinect, in millimetres. Despite being encoded in 16-bits the highest witnessed value measured by the Kinect in different scenes was *9757*, meaning the Kinect cannot see beyond $9,7$ meters of distance, approximately.

The conversion of the image is justified mainly for two reasons: first most of the image-processing algorithms available in the OpenCV framework do not support matrices with encoding larger than 8-bits, and so to reduce development time this approach was preferable; second, processing 16-bits images is computationally more costly than processing 8-bit images, and because this project is meant to be applied in a service robot with other algorithms being employed at the same time, such as navigation and localization, computational cost is an important factor.

In order to preserve the information with the best precision possible when passing from 16-bit images to 8-bit images, the conversion is not applied on the entire original range of $[0, 65536]$ to $[0, 256]$, but rather on a smaller range with greater importance for this application. The larger the range, the further the Kinect will be able to see, but the less precision it will present for the depth values.

To perform the detection of humans, the template matching algorithm needs a complete vision over the person's head and shoulders. This limits the minimum range at about 1 meter from the Kinect, due to the height that the camera is mounted on the CAMBADA@Home and due to the vertical field of view of the camera. As for the maximum range, the further the object is from the camera, the more irregular it's contour will be, and also as stated before the Kinect cannot capture object beyond 9.7, therefore 9 meters was the chosen value for the maximum detection distance.

Given this minimum and maximum distances, the conversion is carried out using Equation 1, creating masks with pixels marked as 0 if the original value of the pixel in the depth image is outside of the range, and 1 if it in included in the range.

$$\mathcal{C}(u,v) = \begin{cases} b\left(\frac{\mathcal{I}(u,v)-\psi\times 1000}{(\gamma-\psi)\times 1000}\right) & \text{, if } \psi < \mathcal{I}(u,v) < \gamma \\ 0 & \text{, otherwise} \end{cases} \tag{1}$$

In Equation 1 $\mathcal{C}$ is the matrix representing the converted image with u rows and v columns, and $\mathcal{I}$ the original image with the same size (640 columns by 480 rows). Variables ψ and γ represent the minimum and maximum depth values in meters respectively, and b the number of bins of the histogram, which in this case is equal to 256 in order to take advantage of the full 8-bit precision.

Histogram analysis Analysing 3D scenes is a computationally heavy task as can be seen by most of the processing time achieved in previous work covered in section 2 of this paper. The method presented in this paper makes use of depth images, instead of point clouds, due to the superior simplicity in data analysis (2D instead of 3D) and use of known image-processing algorithms.

Each pixel in the depth image received from the Kinect ROS driver has a value that represents it's distance in millimetres to the camera. If the image's histogram is calculated, it can effectively demonstrate the most occupied regions in the image. If the objective is then to detect humans in the environment,

these may be considered as continuous regions who occupy a portion of the scene. The objective when analysing the histogram is to search for the most occupied regions, represented by mounds in the histogram, and create slices that encompass these, preferably individually.

Due to the conversion made in the previous stage and the interpolation performed natively by the Kinect for distant points, bins higher than a certain value start to suffer from high variations creating improper local maximums. Therefore, before the detection of the histogram's slices, a median filter is applied only to the bins whose count is equal to 0 to smooth the graph, using Equation 2.

$$\mathcal{H}(b) = \begin{cases} \left(\frac{\mathcal{H}(b-1)+\mathcal{H}(b+1)}{2}\right) & \text{, if } \mathcal{H}(b) = 0 \\ \mathcal{H}(b) & \text{, if } \mathcal{H}(b) > 0 \end{cases} \text{, where } 0 < b < 256 \tag{2}$$

In Equation 2, $\mathcal{H}$ is the image's histogram array and b the number of the bins, where it's value can go from 1 to 256, ignoring bin 0 which is the value assigned to discarded values or pixels outside of the desired conversion range.

The algorithm starts by locating all local maximums, which are characterized by the property represented in Equation 3:

$$H(b-1) < H(b) > H(b+1) \tag{3}$$

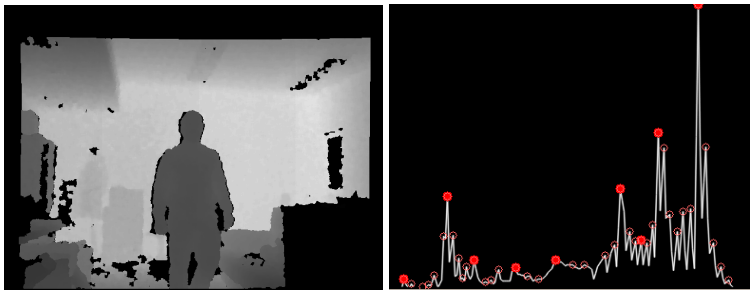
These local maximums represent brightness levels in the scene that are more populated, and if there are any persons (or other objects) in the image, they will most likely be in these levels. Objects in the image have a certain thickness, therefore, determining only the local maximum is not enough, it is necessary to expand these levels into slices that encompass several levels. To optimize the computational cost associated with the processing of each slice, not all maximums will be expanded into a slice, only the most prominent.

In order to obtain only the most important local maximums, Equation 4 is applied to the previously obtained maximums, and discarding those that do not respect that condition, selecting only second order local maximums. The second part of the condition was added to reduce even further the number of local maximums selected for expansion, ignoring those who do not stand out from their neighbours with a count higher than ω.

$$\mathcal{H}(i-1) < \mathcal{H}(i) > \mathcal{H}(i+1) \cap (\mathcal{H}(i) - \mathcal{H}(i-1) > \omega \cup \mathcal{H}(i) - \mathcal{H}(i+1) > \omega) \tag{4}$$

In Figure 2, a capture of the scene is shown in the form of a depth image *(a)* and its corresponding histogram *(b)* after applying Equation 2. In the histogram it is possible to discern several peeks, these are marked by filled circles which were obtained by Equation 4, while outlined circles indicate local maximums obtained by Equation 3.

Obtaining just the maximums of the histogram is not enough to be able to threshold the image, creating the mentioned 2D slices of the 3D image. In the histogram large objects, such as boxes or people, can be seen as mounds. To create the slices of the histogram that encompass these mounds, the second order

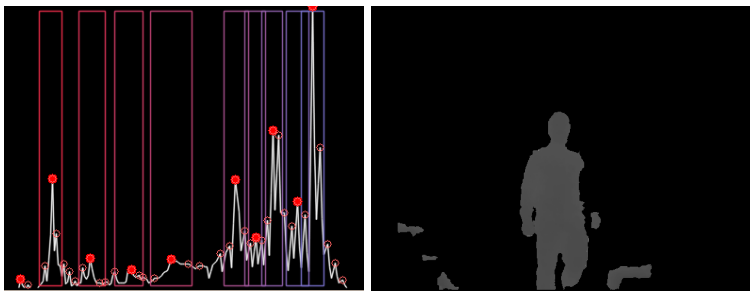


a) b)

Fig. 2. Example of a captured depth image in a), and its corresponding histogram in b).

maximums (red filled circles in Figure 2 *b)*) are taken as starting points, and in the proposed algorithm the slices are expanded to both sides until the base of the mound is reached. The bases of the mound are characterized by the changing of the growth direction. This means that, starting from the local maximum, consecutive bins in both directions should present a decreasing behaviour (hill sides), when this behaviour changes to increasing means that the base of the mound has been reached and the slice is complete.

Histograms perform a pixel count for each brightness level. However there is no information on the position of the pixels. Therefore these slices have to be converted into masks of the real image. This is done by applying a threshold to the image, where pixels that have a brightness encompassed by a given slice of the histogram are marked as 1, and pixels that are not encompassed are marked as 0. This enables the creation of 2D slices of the 3D image, facilitating the detection of contours, discarding unimportant regions, and thereby reducing computational cost associated with the human detection.



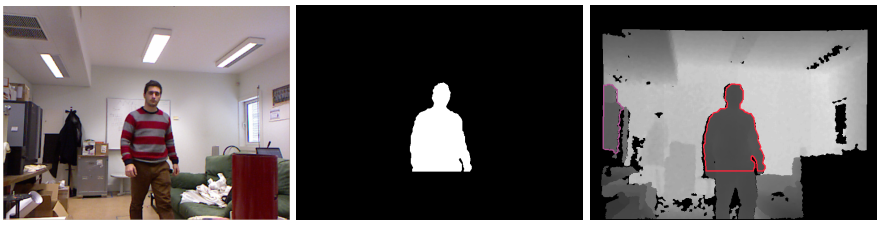
a) b)

Fig. 3. a) Depth image's histogram with slices marked. b) Resulting mask from the first slice

Obtaining Regions of Interest As can be seen in Figure 3, finding the contours of the slices is not enough to create proper ROIs for later classification. A second stage of image-processing is needed in order to separate objects located in the same slice and recover objects that were incorrectly segmented.

First, the contours of isolated objects in each slice are obtained. Next minimum bounding boxes of each contour are calculated, and the centroids of these will be used as a seed pixel for a flood fill algorithm. The lower and upper brightness difference between the seed point and the pixel being flooded are important in order to avoid overflows. These generally happen when two object are in contact, where the most common case, also faced by [4], is when the person's feet are in contact with the ground. In the proposed method the preprocessing applied to the depth image reduces it's precision and does not employ smoothing techniques that are generally time consuming, therefore Xia, L., et al. solution for this problem does not resolve it. our proposal is to cut ROIs at floor height or higher, separating the ground from objects that stand on it.

Using the flood fill method it is possible to restore improperly segmented ROIs during the threshold, as can be seen in Figure 4. Notice how the arm of the person in Figure 3 b) was missing because it was at a different depth from the rest of the body, and how using flood fill recovered the complete region and enabled the separation of unconnected blobs, that were later discarded due to their small size.



a) **b)** **c)**

Fig. 4. Color image capture of the scene in a), Human ROI in b) contour of the ROI overlapped in the depth image in c).

Even before regions are characterized as human or not, it is possible to discard some of the regions obtained. Before being passed to the classification stage presented in subsection 3.2, all the retrieved ROIs are processed by a filter.

This filter is composed by two rules: the first rejects regions based on their proportion and the second based on their occupied area. The first rule will not allow regions to have a bounding box with proportion bigger than 1.7. The proportion of a bounding can be obtained by diving its width by the height, and human physiology dictates that the difference between a persons arm span and it's height is of a few centimetres Therefore even with the arms wide open a person's proportion should not go beyond 1.0 to 1.2. However, to compensate

for occlusion, this limit is extended up to 1.7. The second rule is that a region cannot have an area bigger than $(96882 + \beta) \times e^{(-0.561 \times d)}$, where d is the ROIs average depth and β is used as a control value to increase this limit for finner-tunning because a persons size may differ greatly. The equation was obtained using measures taken from a dataset recorded in our lab with few subjects. In both cases, the limits for both proportion and area, should not be very strict because it is preferable for a non-human ROI to pass through this phase, than to incorrectly eliminate human ROIs

3.2 Information Fusion and Analysis

In the previous stage, regions of interest were retrieved from the depth image, however these were not classified. The classification is performed in this stage, and to do so, information is drawn from the depth and color image.

Depth information is only capable of delivering information on shapes, but because the human shape can differ greatly, tracking is performed on the color image. At this point of the project only depth information is used to classify a ROI, however, other methods are being studied to reduce the number of false positives generated by the template matching algorithm, such as the use of a thermal imaging.

Given the ROIs obtained previously, these will be classified as human or not through a template matching algorithm. The RPROP algorithm [11] was chosen due to its effectiveness and low computational cost unlike most template matching algorithms which perform rastering on the complete image.

In order to keep track of the humans individual position during their presence in the field of view of the camera, a histogram comparison method is used allowing for the same region to be related across consecutive frames, obtaining information from the color image.

Human Classification When using template matching techniques, the choice of the template is crucial for good results, and because the human body presents several degrees of freedom, the choice for which shape to test is done considering the less deformable area. Researchers, such as Krishnamurthy, Su. [5],state that a head-shoulder template, also associated with the Ω shape, is the best template to use when trying to detect humans because this is the less deformable part of our body.

Template matching algorithms usually work by sliding a template across an image and calculating an error for the match between the template, and the image being tested. This is a computationally demanding method that requires a lot of processing power if it is intended to run in real-time.

Researchers such as [4] and [7] use a *Image Pyramid* to perform the template matching for objects with different sizes, due to the perspective effect of monocular cameras caused by the distance of the object relative to the camera. The original image is considered to be the base of the pyramid and at each level the image is downsized. This allows for the same template to be used for detection at different distances.

In our approach the template itself is resized according to the person's distance to the camera and its tested only over the ROIs, not the entire image. This allows for a single template to be used for detection at all distances with just one test for each ROI, instead of n tests according to the number of levels of the pyramid. The template is resized according to Equation 5, which was obtained by fitting the template manually on ROIs classified as human, obtained from our dataset, and obtaining the scale factor necessary for a perfect fit.

$$s = -52.6 \times log(d) + 130.06 \tag{5}$$

In Equation 5, s represents the scale factor by which the template will be multiplied, and d is the average depth of the ROI in meters.

The proposed classification method uses part of a the Perfect Match localization algorithm [12], in which the RPROP algorithm [11] searches for the minimal matching error, using a gradient descent technique, in order to fit the template over the ROI's contour. A *Distance Transform* (DT) map is created from the contour of the ROI and the template is then transversed through it, impelled towards the direction that generates the less error. This algorithm is capable of finding a local minimum error for a particular position of the template over the ROI. This position will indicate the center of the head of the person, if it is indeed a human object, with an associated error.

Depending on the starting position, the final and possibly best position of the template can be achieved in 15 iterations, revealing the localization of the person's head. In this particular case of human detection, where humans are assumed to be in an upright position, the algorithm calculates the start position by dividing the ROI's contour in a predefined number of vertical sections, and chooses the one that is most occupied, for example if the person has one arm stretched the head will not be in the middle section of the ROI, but instead more to the left or to the right depending on the arm stretched.

Choosing the best start position is very important due to the susceptibility of the algorithm converging to an incorrect local minimum. Figure 5 presents a correct match in b) and an incorrect match in d) due to a local minimum located between the arm and the head.

The classification as human or not is performed by judging the error for the best position. If this error is bellow a certain threshold the region is considered human, if it is above or equal than the region is considered not-human.

Tracking Human Regions In order to keep track of the position of the same person throughout their presence in the scene, it is necessary to relate the same region across consecutive frames. Again, because the human body can shift its shape considerably, depth image is a poor choice when it comes to characterize regions. Consequently tracking relies on the color image provided by the Kinect camera.

A proven detection method is Mean Shift applied to image analysis, in which a particular object is located in a back projection image. Back projection uses

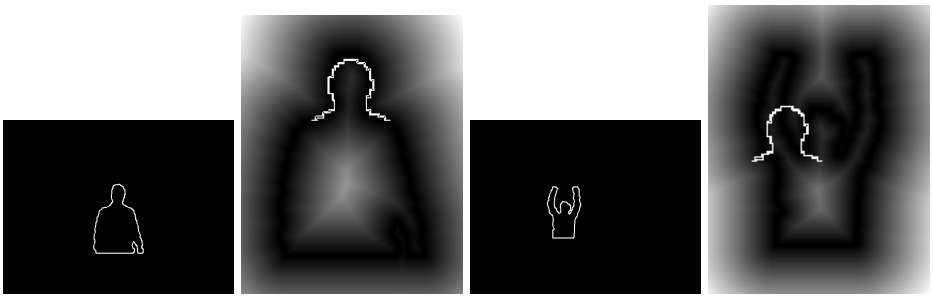


Fig. 5. Example of a correct and incorrect match location. a) and c) present the contour of the ROI, and b) and d) the resulting DT map with the overlapped template.

the histogram of the desired object as it's feature and is then capable of creating an array of statistical probability for the location of that object in a search area.

However, at this point, in addition to the human object contour, its position on the whole image is also known. Therefore, it is not a question of searching for the same object in consecutive frames, but instead to determine whose regions present in the previous frame are still present in the current frame, if any.

Inspired by the method used in the Mean Shift algorithm, the proposed tracking method uses histogram comparison to relate regions across image frames. The HSV color space was chosen for the comparison due to its flexibility when representing colors using only the Hue channel, and its greater robustness to changes in lighting conditions when compared to other color spaces. This is important when tracking a person across different house divisions, for example, where the lighting conditions may change drastically due to different light sources such as windows or lamps, or even shadows cast by large objects or walls.

The proposed method generates a histogram for the portion of the color image that is masked by ROIs that were classified as human, and normalizes it so that the area occupied by the region is not important but instead the percentage of each color in the ROI. The histograms from the currently tracked ROIs are then compared to previous tracked ROIs in order to calculate a difference value. This difference can be seen as an error that is computed by comparing the value of each histogram's bins, and summing the difference for each channel.

After comparing each current human ROI with previously tracked human ROIs stored in memory, it is necessary to associate them without repetitions. Choosing which current region is assigned to which previous frame's region can be then seen as an optimal assignment problem.

The *Hungarian algorithm* was studied as a possible solution for this problem. However this method did not present a usable solution due to it's restrictions. Therefore a new optimal assignment method is proposed where each human ROI selects the region present in the previous frame with the least error, and if two or more ROIs select the same tracked region, only the one with the lowest error

will maintain it's choice while others will forfeit it and choose a different one, until all regions have chosen different previously tracked regions.

In addition, to ensure that each human object is paired with its equivalent in consecutive frames, the algorithm must also recognize when a error, although being the lowest, is still to high to guarantee that the region is in fact the same, therefore errors higher than a certain value are considered new ROIs.

4 Final Remarks

In this paper a new method for people detection and tracking is proposed, inspired by recent work in this area, but with modifications that allow for a reduced computational cost when compared to other solution that use 3D information.

The method employed for ROIs detection presents satisfactory preliminary results, both in the form of detection rates and computational cost. Slicing the 3D scene in 2D images enables the use of know image analysis techniques while at the same time proves to be a lightweight process in term of computational cost.

The classification phase of the proposed method uses a template matching technique aided by the RPROP gradient descent algorithm, a first use of this algorithm in a template matching technique as far as the author's concern. Due to the reduced area of the ROIs when compared to the whole image, the algorithm is able to find a solution in 15 iterations and return a position with a local minimum error, which in cases where the head and shoulders are minimally visible is usually the correct location.

By adjusting the minimum error necessary for a region to be considered a human, it is possible to reduce the number of detections classified as human incorrectly, while increasing this threshold value causes human objects to be classified as non-human. Further study is needed in order to determine another classification method that is able to complement these disadvantages, possibly using the color image or even thermal images. Also, because only one template is used for now, people facing sideways to the camera are not properly classified. However, judging by the results of front and back facing people detection, if more templates are considered this problem can be solved without increasing processing time considerably.

Finally, tracking through histogram comparison and association of identical human ROIs across consecutive frames has proven to be effective in most cases, being able to associate the same person during its entire presence in the scene and even after it disappears and reappears, whether is due to a bad detection in the ROI retrieving phase or simply because the person stepped out of the camera's field of vision. Nevertheless, more tests have to be carried out in order to determine the color space and individual channel's error weight that maximize the difference between the smallest error and the rest.

The pipeline in mainly divided in two stages, the object detection phase and the classification phase. Object detection is able to extract all ROIs in an image in an average of 48 miliseconds, while object classification and tracking is performed

at an average of 54 miliseconds. Because the main stages are implemented in different ROS nodes, the detection stage can be analysing frame i while the classification stage is presenting the results for frame $i-1$, this makes the system capable of functioning at about 15-20 frames per second in a mid-range laptop with an Intel Core i5 processor, depending on the entropy of the environment.

5 Acknowledgments

This research is funded by FEDER through the Operational Program Competitiveness Factors - COMPETE and by National Funds through FCT - Foundation for Science and Technology in the context of the project FCOMP-01-0124-FEDER-022682 (FCT reference PEst-C/EEI/UI0127/2011)

References

1. I. Haritaoglu, D. Harwood, and L.S. Davis. W4: real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):809–830, 2000.
2. G.L. Foresti, L. Marcenaro, and C.S. Regazzoni. Automatic detection and indexing of video-event shots for surveillance applications. *Multimedia, IEEE Transactions on*, 4(4):459–471, 2002.
3. F. Guan, L. Li, S. Ge, and A. Loh. Robust human detection and identification by using stereo and thermal images in human robot interaction. *International Journal of Information Acquisition*, 04(02):161–183, 2007.
4. Lu Xia, Chia-Chih Chen, and J.K. Aggarwal. Human detection using depth information by kinect. pages 15–22, 2011.
5. Su. Krishnamurthy. Human detection and extraction using kinect depth images. *www1bpt.bridgeport.edu*, 2011.
6. Sho Ikemura and Hironobu Fujiyoshi. Real-time human detection using relational depth similarity features. *Computer VisionACCV 2010*, pages 1–14, 2011.
7. J. Davis and M. Keck. A two-stage template approach to person detection in thermal imagery. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 364–369, 2005.
8. Mauricio Correa, Gabriel Hermosilla, Rodrigo Verschae, and Javier Ruiz-del Solar. Human Detection and Identification by Robots Using Thermal and Visual Information in Domestic Environments. *Journal of Intelligent I& Robotic Systems*, 66(1-2):223–243, July 2011.
9. L. Spinello and K. Arras. People detection in RGB-D data. pages 3838–3843, 2011.
10. Frederik Hegger and Nico Hochgeschwender. People Detection in 3D Point Clouds using Local Surface Normals. *ais.uni-bonn.de*, 2011.
11. M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591 vol.1, 1993.
12. Martin Lauer, Sascha Lange, and Martin Riedmiller. Calculating the perfect match: An efficient and accurate approach for robot self-localization. In *in RoboCup Symposium*, pages 142–153. Springer Verlag, 2005.