Image and Vision Computing 28 (2010) 391–402

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Human detection using a mobile platform and novel features derived from a visual saliency mechanism

Sebastian Montabone *, Alvaro Soto

Department of Computer Science, Pontificia Universidad Catolica de Chile, Casilla 306, Santiago 22, Chile

ARTICLE INFO

Article history:
Received 7 July 2008
Received in revised form 9 January 2009
Accepted 7 June 2009

Keywords:
Human detection
Visual saliency
Visual features
Moving cameras

ABSTRACT

Human detection is a key ability to an increasing number of applications that operates in human inhabited environments or needs to interact with a human user. Currently, most successful approaches to human detection are based on background substraction techniques that apply only to the case of static cameras or cameras with highly constrained motions. Furthermore, many applications rely on features derived from specific human poses, such as systems based on features derived from the human face which is only visible when a person is facing the detecting camera. In this work, we present a new computer vision algorithm designed to operate with moving cameras and to detect humans in different poses under partial or complete view of the human body. We follow a standard pattern recognition approach based on four main steps: (i) preprocessing to achieve color constancy and stereo pair calibration, (ii) segmentation using depth continuity information, (iii) feature extraction based on visual saliency, and (iv) classification using a neural network. The main novelty of our approach lies in the feature extraction step, where we propose novel features derived from a visual saliency mechanism. In contrast to previous works, we do not use a pyramidal decomposition to run the saliency algorithm, but we implement this at the original image resolution using the so-called integral image. Our results indicate that our method: (i) outperforms state-of-the-art techniques for human detection based on face detectors, (ii) outperforms state-of-the-art techniques for complete human body detection based on different set of visual features, and (iii) operates in real time onboard a mobile platform, such as a mobile robot (15 fps).

1. Introduction

Human detection is a key ability to an increasing number of applications that operates in human inhabited environments or needs to interact with a human user. As an example, cars provided with pedestrian protection systems require some type of human detection capability [12]. In the same way, an autonomous mobile robot, such as a social robot [8], needs to detect humans to perform tasks like aid for rehabilitation in hospitals [21], assistance in offices [1], or guidance in museums [4].

There is an extensive list of works dedicated to the problem of human detection [24]. Most of these works are based on static cameras, where the goal is not only to detect but also to track humans [39]. In this case, the most popular approach is to detect humans using background subtraction methods. These methods require that the image background does not change severely between frames. Systems using background subtraction first build a background model of the controlled environment. This has been done in many different ways [24]; one example is generating a Gaussian distribution of the background as seen in [39]. When the platform is mobile, however, as in the case of a mobile robot, none of the background subtraction methods can be applied.

An important aspect of human detection is that the system should be able to detect a human under different poses, however, most current solutions still rely on detecting human faces [3,14,23,20]. This has the disadvantage that a human can only be detected when is facing the camera. In particular, in the case of a mobile robot this leads to a loss of several social aspects. For instance, if the robot wants to initiate a conversation, the user has to already be paying attention to it, which is not always the case. Similar problems might arise in the case of a pedestrian detection system where the human can be in any pose with respect to the detecting system.

In this way, although important advances have been achieved in the development of algorithms for human detection, there is still space for further improvements, particularly in the case of a mobile platform that needs to detect humans under different poses [13]. Furthermore, many applications also require real time operation, stressing the need for an efficient human detection system that can provide a timely detection. These are the main challenges that we face in this work.

In this work, we present a human detection system able to: (i) operate on a mobile platform, (ii) detect humans under different

* Corresponding author. Tel.: +56 2 6864440; fax: +56 2 3544444.
E-mail addresses: samontab@puc.cl (S. Montabone), asoto@ing.puc.cl (A. Soto).

0262-8856/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.06.006

poses, including frontal, back, and profile views of the complete and upper human body, and (iii) operate in real time (15 fps). The key principle that provides such flexibility in the detection is the use of novel features for human detection derived from a visual saliency mechanism. We call these features: visual saliency features or VSFs, and we refer to the method to extract them as VSF. These features are based on a biologically inspired attention system. Due to this, the size and shape of the filters used for calculations of the proposed features are not user defined like most previously used features for human detection [27,34]. Experiments in real life scenarios show that the proposed system: (i) outperforms state-of-the-art techniques for human detection based on face detectors, (ii) outperforms related systems based on different set of visual features, and (iii) operates in real time onboard a mobile robot.

This paper is organized as follows. Section 2 reviews relevant previous work on human detection using computer vision techniques. Section 3 presents relevant background material. Section 4 discusses the main details of our approach. Section 5 shows the results of testing our algorithm under different real case scenarios. Finally, Section 6 presents the main conclusions of this work.

2. Previous work

The state-of-the-art in human detection systems can be divided into two main categories: (i) Methods that require background substraction as a first step to detect the interesting objects. (ii) Methods that perform the detection using uncontrolled moving cameras. Our method belongs to this last category, hence, we concentrate our review in methods that do not rely in background subtraction techniques. For a more extensive review refer to [13,11].

In 2004, Gavrila et al. proposed a pedestrian protection system for moving vehicles [12]. Human bodies are detected using a shape-based method known as the Chamfer system. Every detected shape is then passed to a previously trained neural network as a verification step using texture as the feature. As stated by the authors, this system requires more accuracy for use in the real world.

Rajagopalan and Chellappa proposed an statistical approach for describing the shape of humans using clusters [29] and later generalized it for detecting humans and vehicles [30].

In 2000, Zhao and Thorpe presented an interesting work where human detection is accomplished using the complete human body [40]. It uses a neural network fed with the intensity gradient of the objects. Stereo information is used for removing areas of the image that do not belong to the object itself in order to diminish the negative effects of background clutter. The reported detection rate of this system is 85.4%.

In the work proposed by Papageorgiou and Poggio, human detection is performed using Haar-like features over a previously trained support vector machine classifier [27]. Haar-like features are intensity differences at user defined rectangular regions. The size of these rectangular regions is object dependent. The authors manually select this parameter on their training data obtaining a detection rate of 90%.

In 2001, Viola and Jones constructed a fast frontal face detection system based on an extended set of Haar-like features and an AdaBoost classifier [34]. It also presents the use of an integral image in order to speed up the feature calculation process (Ref. Section 3.1). This system was later used in many works due to its real time operation and accuracy [14,3,23,37]. Also, this system was later updated to use motion in order to increase the detection rates [35].

Color cues have also being used to detect humans, mainly motivated by the robustness of skin color detection techniques [33]. In [28], Pszczolkowski and Soto present a human detection system for a mobile platform based on depth segmentation and a new color based technique to detect faces. In [23], Munoz-Salinas et al. also use color and depth information for human detection by combining plan-view map information and a color face detector.

3. Background information

In this section, first the concept of an integral image is explained. Afterwards, attention systems and saliency maps are presented. Finally, computational models of visual saliency are introduced and compared.

3.1. Integral image

This concept was first referred to as summed-area tables or texture mapping in the field of Computer Graphics [5]. Later, this idea was brought to image processing in the work of [34]. They presented a revolutionary investigation in the object detection area, producing results up to 15 times faster than previous works.

Given a grayscale image i, each pair (x, y) of the integral image I of i represents the sum of the image values above and to the left of x, y:

$$I(x,y) = \sum_{x' \leqslant x, y' \leqslant y} i(x', y'). \tag{1}$$

Therefore, given any particular rectangular area defined by $P1 = (x1, y1)$ and $P2 = (x2, y2)$ its sum can be calculated in constant time using the integral image:

$$rectSum(x1, y1, x2, y2) = I(x2, y2) - I(x1, y2) - I(x2, y1) + I(x1, y1). \tag{2}$$

3.2. Attention systems and saliency maps

Attention systems are used to compute interesting areas of a given image or video. This is done because of the vast amounts of information that a computer vision system normally needs to process. It is impractical for those systems to exhaustively search through all the data. Therefore, only the interesting regions computed by an attention system are used, considerably decrementing the complexity of the tasks.

The most common attention systems are based on biological models [32]. These models suggest that the human visual system uses only a portion of the received information in order to achieve faster results when dealing with complex scenes. This portion is known as the Focus of Attention. One of the most accepted theories about the Focus of Attention is the Treisman's Feature-integration Theory of Attention [32]. Basically, he proposes that a saliency map is built by mixing parallel feature maps.

The retina of the human eye contains ganglion cells which receive the visual information from photo receptors through the bipolar cells. The receptive fields of the ganglion cells are composed of two areas, the surround and the center. There are two types of ganglion cells: (i) on-center ganglion cells which respond to bright areas surrounded by a dark background, and (ii) off-center ganglion cells which respond to dark areas surrounded by a bright background (Fig. 1) [26]. Most attention systems build upon the base of the Theory of Attention using center–surround differences [17,9,32].

3.3. Computational models of visual saliency

Computational models of visual saliency often try to mimic the behavior of the receptive fields of ganglion cells. Because of calculations of on-center and off-center differences, expensive computa-

tions needs to be done. Commonly, there is a trade-off between accuracy and speed of computation. As a result, usually attention systems only deliver coarse grained information to decrease processing time.

One of the most accepted and widely used computational models of attention was proposed by Itti et al. [17]. This system has a solid theoretical background, is able to integrate different visual cues, [32] and performs relatively well. In addition to this, a completely documented and supported implementation of the system, the iLab Neuromorphic Vision C++ Toolkit (iNVT), is publicly available [16].

Over the past few years, the original work of [17] was improved in the system called VOCUS proposed by Frintrop [9]. Retaining the same theory from the work of [17], but implementing it in a different manner, Frintrop managed to deliver more accurate results, but at the expense of more computational time.

The center–surround differences calculated by iNVT and VOCUS are slow. To increase the speed of the systems, both works adopted two approximations: (i) squared regions, and (ii) image pyramidal decomposition. In terms of squared regions, center–surround calculations in the human eye are circular [26]. For simplicity, iNVT and VOCUS approximate these regions with squares. No substantial difference is presented in the results when using circular areas instead of squared ones [10]. In terms of pyramidal decomposition, as the system works only with the most downsampled scales in the pyramid, the output quality gets compromised. This normally leads to poorly defined borders of the objects in the resulting saliency map. Although VOCUS uses the same concept of image pyramids as iNVT, the particular method used by Frintrop yields better results [9].

iNVT and VOCUS use center–surround differences to calculate saliency maps based on different visual cues. For simplicity, only intensity map computation will be described next for each system. The use of center–surround differences for the calculation of other saliency maps is analogous.

3.3.1. iNVT

This system first generates a grayscale version of the input image. Then, it calculates an image pyramid of eight grayscale images, each one of them scaled to one quarter of the previous one.

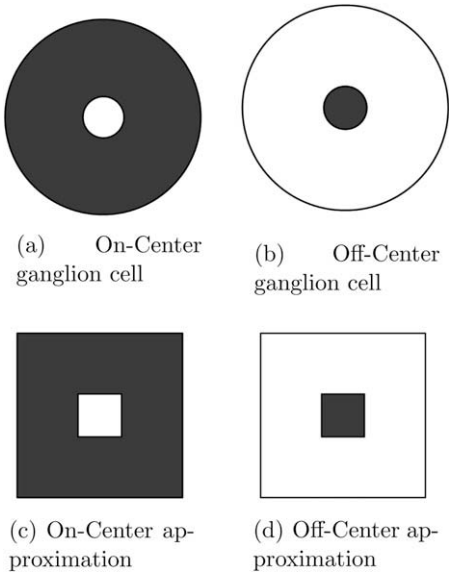


Fig. 1. On-center and off-center ganglion cells and their approximation on computational models of visual saliency.

After that, center–surround differences are calculated as an across-scale difference between coarse and fine scales. Fine scales are defined as a scale $s \in \{2,3,4\}$ and coarse scales are defined as a scale $s \in \{5,6,7,8\}$.

An across-scale difference is calculated by scaling the coarse scale into the fine scale and then executing pixel by pixel subtraction. Next, all the maps are summed up to obtain the final intensity map. This yields fast but very poor feature maps [17].

Also, center–surround differences are calculated as absolute values. In other words, no difference exists between on-center and off-center differences for this system. This can be calculated faster, but lacks the flexibility of separated maps [9].

3.3.2. Vocus

In VOCUS [9], the intensity map is calculated as follows: first, the original color image is converted into grayscale. Then, a Gaussian image pyramid is created. This is achieved by applying a 3×3 Gaussian filter to the grayscale image, and after that, scaling it down by a factor of two on each axis. The filtering and scaling are repeated four times, yielding five images: i_0, i_1, i_2, i_3, and i_4. From this moment on, the system only takes into account the information present in the smallest scales; images i_2, i_3, and i_4 (see Fig. 2).

The system now calculates on-center and off-center differences in the three images that represent scales $s \in \{2,3,4\}$, respectively. Centers are represented as a pixel and two surround values, σ, are used: 3 and 7, based in the work of [17]. Therefore, 12 intensity submaps are generated. The process of calculating these submaps is as follows: first, center and surround are defined:

$$surround(x,y,s,\sigma) = \frac{\sum_{x'=-\sigma}^{x'=\sigma}\sum_{y'=-\sigma}^{y'=\sigma} i_s(x+x', y+y') - i_s(x,y)}{(2\sigma+1)^2 - 1}, \quad (3)$$

$$center(x,y,s) = i_s(x,y), \quad (4)$$

then, every pixel of each intensity submap is calculated:

$$Int_{On,s,\sigma}(x,y) = \max\{center(x,y,s) - surround(x,y,s,\sigma), 0\}, \quad (5)$$

$$Int_{Off,s,\sigma}(x,y) = \max\{surround(x,y,s,\sigma) - center(x,y,s), 0\}, \quad (6)$$

where $s \in \{2,3,4\}$ represents the image scale, $\sigma \in \{3,7\}$ the surround, and On, Off, the on-center and off-center differences, respectively.

After that, an on-center intensity map is calculated. This is done scaling the six on-center intensity submaps into the largest scale; i_2, and then summing pixel by pixel. An off-center intensity map is generated the same way, using the off-center submaps.

$$Int_{On} = \oplus_{s,\sigma} Int_{On,s,\sigma}, \quad (7)$$

$$Int_{Off} = \oplus_{s,\sigma} Int_{Off,s,\sigma}, \quad (8)$$

where $\oplus$ denotes the across-scale sum previously explained.

3.3.3. Discussion

The are two main differences between VOCUS and iNVT: The first one is that VOCUS generates independent on-center and off-center intensity maps, whereas iNVT only calculates the absolute difference between center and surround. This gives VOCUS the advantage to distinguish between these two features [9].

The other main difference is that when iNVT calculates the center surround differences, it subtracts images from fine and coarse scales, resizing to the finest scale, therefore yielding less defined borders. VOCUS instead, first calculates center surround differences in every scaled image and then resizes the images to the largest scale, adding all the computed intensity submaps pixel by pixel. This technique yields better results than the work of [17] but the processing time is slower [9].

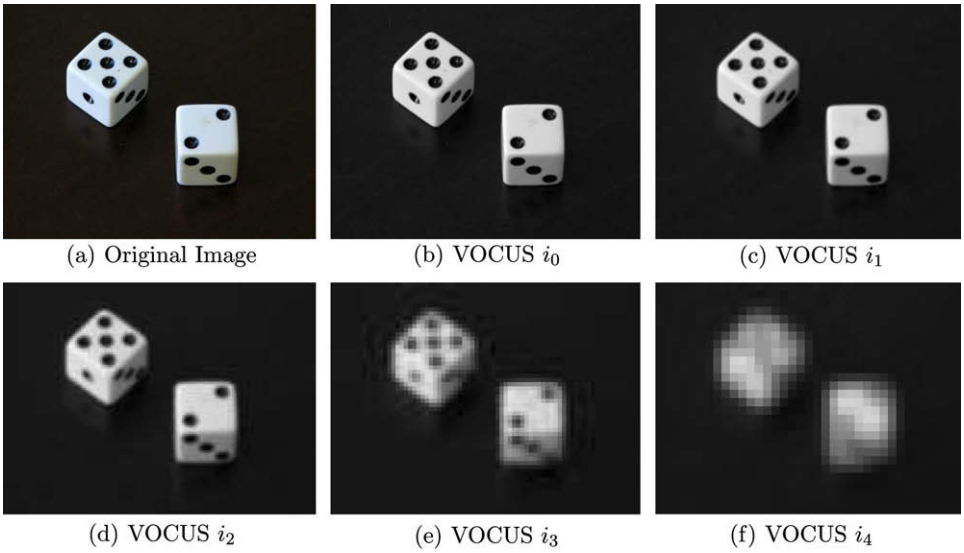


Fig. 2. VOCUS image scales. In VOCUS, the original image is converted into grayscale. Next, five different image scales are created $(i_0, \ldots, i_4)$. The system then works with the smallest scales (i_2, i_3, i_4), represented in the bottom images.

3.3.4. Drawbacks of iNVT and VOCUS

The main drawback of iNVT and VOCUS is that the resulting feature maps are very poor. They do not contain the same level of resolution provided by the original image. These works first scale down the image by a factor of 16 resulting in considerable detail loss. Then, they proceed to build an image pyramidal decomposition, further decreasing the details.

Generating good quality feature maps is needed in order to obtain fine grained information of the objects. Until now, all attention systems only provide coarse grained information in their feature maps. This is the deficiency that we exploit to use saliency information as a novel feature descriptor mechanism to directly achieve classification instead of only as a visual attention mechanism [36,17,9,7] or a relevant region detector [18,22,31] as in previous works.

4. Our approach

Our approach for human detection follows a standard pattern recognition scheme based on four main steps: (i) preprocessing for stereo pair calibration, (ii) segmentation using depth continuity information, (iii) Feature extraction based on visual saliency, and (iv) classification using a neural network. An overview of the overall approach is shown in Fig. 3. We describe next the details of each step.

4.1. Preprocessing module

4.1.1. Rectification

The original images taken from the stereo camera are first rectified using the calibration parameters obtained using the small vi-

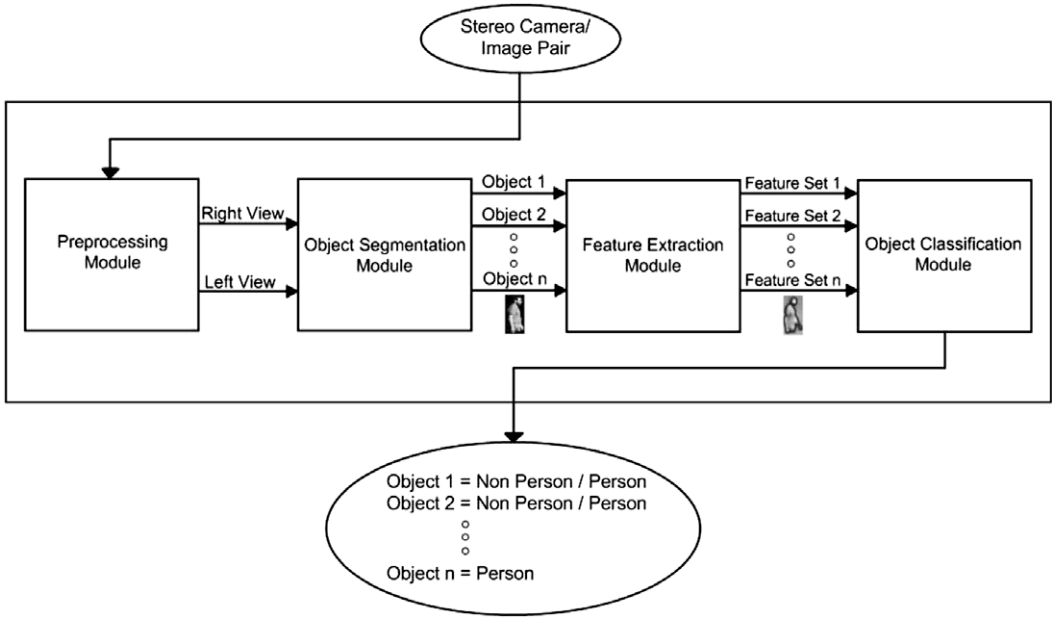


Fig. 3. General diagram of the proposed system.

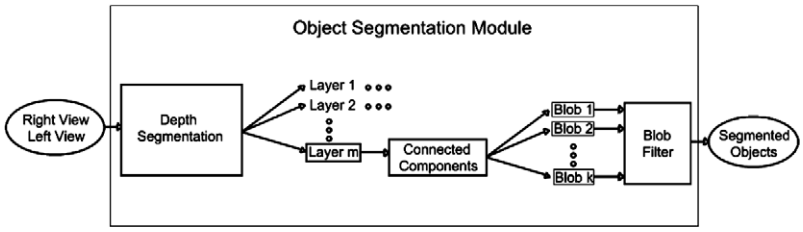


Fig. 4. Object segmentation module diagram. First, layers are obtained through depth segmentation using stereo analysis. Then, connected components analysis is used in order to extract blobs from each layer. Each blob is then filtered using size and shape constraints.

sion system (SVS) [19] calibration routine. A standard chessboard printed image was used to calibrate the device.

4.2. Object segmentation module

Stereo information obtained from the SVS is used to segment objects from the scene. The system uses a segmentation approach based on the work of [15]. There are three steps involved in this module: (i) depth segmentation, (ii) analysis of connected components on each previously segmented layer, and (iii) application of set of filters in cascade. A diagram of this module can be seen in Fig. 4. The details of the different steps are presented next.

First, the system calculates the disparity image for the entire frame and a disparity histogram is generated. Each local maxima of the disparity image represents the existence of one or more solid objects at a particular depth. The original depth image is segmented into i layers, where i is the number of local maxima in the histogram of disparities. Closer objects often present more variable depth values than distant ones. When extracting the ith layer from the depth map a depth dependent value Δd_i is generated first. This value is calculated using a polynomial fit on previously obtained correct human segmentation values at different depths. Only values that lie inside the depth range defined by $(d_i - \Delta d_i, d_i + \Delta d_i)$ are present in the ith layer. Every layer is separated from the original depth map generating isolated images of different objects at different distances. Afterwards, every layer is segmented using standard connected components analysis.

As a final post-processing step, every blob is passed through a cascade set of filters. The system filters out candidate blobs using size and shape constraints. An estimate of the object's real height, width and depth can be obtained using stereo vision. Also, the bounding box that defines each blob provides aspect ratio information. In this way, using thresholds adapted to the expected dimensions of a human in an image, the system filters out blobs that: (i) present a real world height that is less than 40 cm, (ii) are wider than its height, or (iii) have a height that is more than three times its width. Filtered out blobs are split using a simple method. This method consists in measuring the valid height for every line in the blob, yielding a maximum valid height. Every line that is shorter than half of the maximum height is eliminated, splitting the blob.

It is worth to mention that the previous scheme provides a high level of freedom to move the video cameras, as opposed to other segmentation methods commonly used to detect human by a mobile robot, such as methods based on floor plane assumptions [15].

4.3. Feature extraction module

Attention systems are commonly used in computer vision applications as preprocessing modules (Ref. Section 3.2). As we explained before, the high computational cost of running visual attention systems at full image resolution has avoided the use of saliency mechanisms as direct feature extraction methods. This explains why most common attention systems provide only coarse grained information.

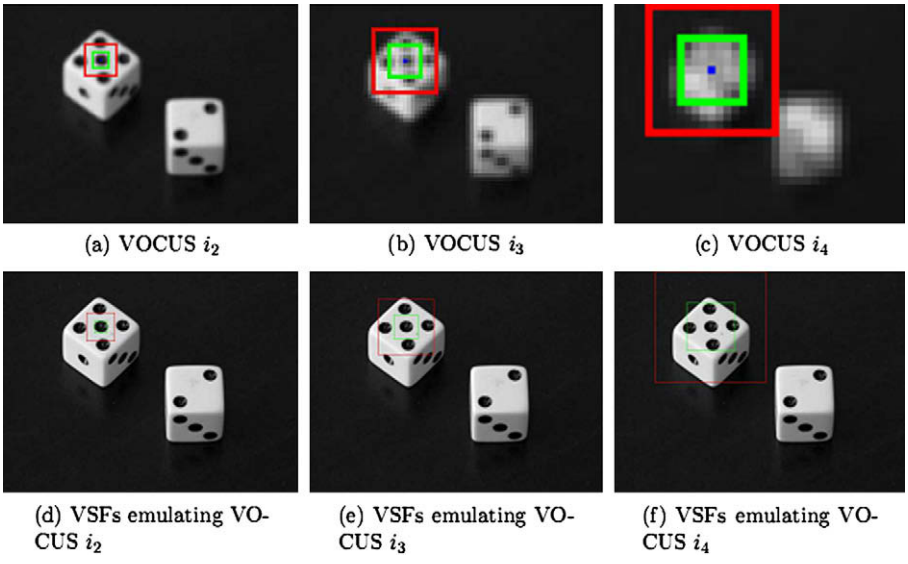


Fig. 5. Top: filter windows used in VOCUS for scales 2, 3, and 4, respectively. Red windows (larger) represent $\sigma = 7$, and green (smaller), $\sigma = 3$. Bottom: filter windows used in VSF, red windows represent ς values of 28, 56, and 112, respectively, while green windows represent ς values of 12, 24, and 48, respectively. Notice how the filters in VOCUS loose detail as the size of the filter window grows larger, in contrast, the ones in VSF preserve all their details at all scales.

This work proposes the use of visual saliency as direct features for human detection. In order to obtain high quality features, a novel way of calculating visual saliency is presented. The proposed features or VSFs use the same biological theory as [9,17] but uses an integral image on the original scale in order to obtain high quality features in real time (Ref. Section 3.1).

First, a Gaussian filter with a 3×3 window is used twice, in order to smooth the image and obtain the same robustness to noise as the previous works [9,17]. Then, the system calculates on-center and off-center differences separately using a unique integral image with variable size filter windows over the original grayscale image (see Fig. 5).

4.3.1. Filter windows

VOCUS filter windows are defined by the scale $s \in \{2,3,4\}$ and the surround $\sigma \in \{3,7\}$. Therefore there are six different sized filter windows in VOCUS. Using them in order to calculate on-center and off-center differences yield the 12 intensity submaps previously referred to.

The VSF method implements all of the same filter windows used in VOCUS. The main difference is that the filter is applied to the entire original image, instead of scaled down versions. Therefore, the system uses only a single parameter to define all the filter windows that will be calculated on a single integral image:

$$\varsigma = \sigma 2^s. \tag{9}$$

where σ represents the surround and s, the scale, both used in the VOCUS system. Also, ς denotes the surround to be used in VSF in order to cover the same window as the VOCUS window.

4.3.2. Feature calculation

In order to calculate the intensity submaps, we first define center and surround using Eq. (2):

$$surround(x,y,\varsigma) = \frac{rectSum(x-\varsigma, y-\varsigma, x+\varsigma, y+\varsigma) - i(x,y)}{(2\varsigma+1)^2 - 1}, \tag{10}$$

$$center(x,y) = i(x,y). \tag{11}$$

Then, every pixel of each intensity submap is calculated as follows:

$$Int_{On,\varsigma}(x,y) = \max\{center(x,y) - surround(x,y,\varsigma), 0\}, \tag{12}$$

$$Int_{Off,\varsigma}(x,y) = \max\{surround(x,y,\varsigma) - center(x,y), 0\}, \tag{13}$$

where $\varsigma \in \{12,24,28,48,56,112\}$ represents the surround, and *On*, *Off*, the on-center and off-center differences, respectively. Note that the values of ς are specially calculated using Eq. (9) in order to process the same windows as the VOCUS system (see Fig. 5).

Then, an on-center intensity map is calculated. This is done summing the six on-center intensity submaps pixel by pixel. An off-center intensity map is generated the same way, using the off-center submaps. Finally, both maps are summed up.

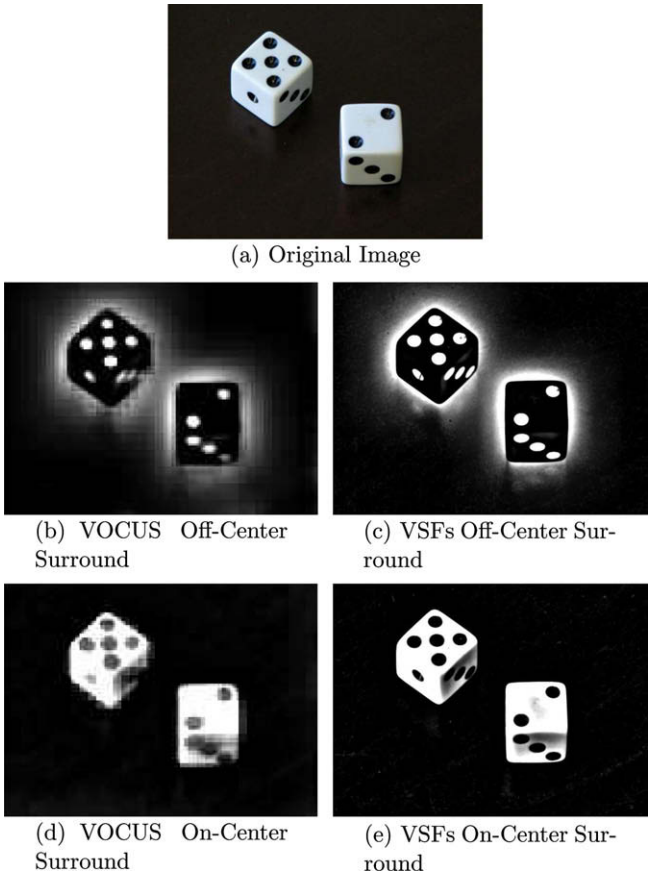
$$Int_{On} = \sum_{\varsigma} Int_{On,\varsigma}, \tag{14}$$

$$Int_{Off} = \sum_{\varsigma} Int_{Off,\varsigma}. \tag{15}$$

All the details of the image are preserved because the surround window varies according to the surrounding and scaling values proposed in VOCUS. This is done in order to calculate the same center–surround differences but in a highly accurate way.

4.3.3. Center–surround calculation results

The results shown in Figs. 6 and 7 demonstrate the positive effects of not scaling the images when calculating the center–surround differences. VSF provides fine grained feature maps and much more defined borders. Other systems, such as VOCUS, only generate coarse grained feature maps with poorly defined borders.


(a) Original Image

(b) VOCUS Off-Center Surround

(c) VSFs Off-Center Surround

(d) VOCUS On-Center Surround

(e) VSFs On-Center Surround

Fig. 6. Comparing results of VOCUS and VSF. Notice how the detail is preserved in VSF (right) and how the VOCUS results are poorly defined (left).

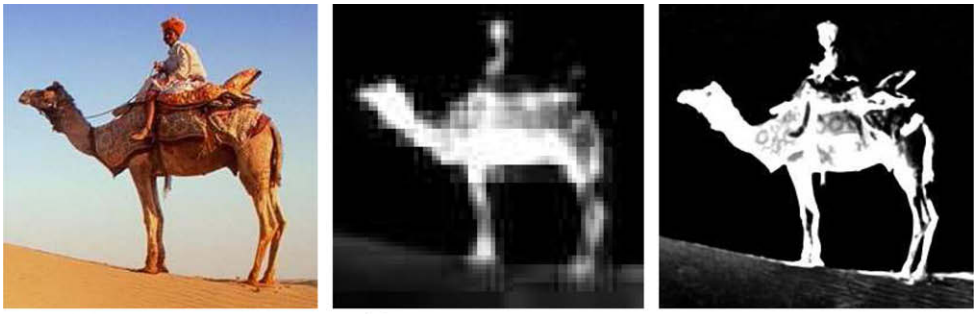
In this work, a novel feature set is proposed: VSFs. Each segmented object is passed to the feature extracting module, obtaining VSFs. These VSFs represent the most salient parts of each object. As these features are biologically inspired, filters size and shape are obtained from previous investigations on how the retina perceives light stimuli [26] instead of being user defined as most previously used features for human detection. Examples of VSFs can be seen in Fig. 8.

4.3.4. VSF vs VOCUS computational costs

VOCUS explicitly uses Eq. (4) to determine its features for each image pixel at every scale. Given a center–surround filter window of n^2 pixels, VOCUS feature calculation has order $\mathcal{O}(n^2)$ for each filter because it sums every term individually. On the other hand, VSF uses the integral image (Ref. Section 3.1) to obtain the same calculation in constant time ($\mathcal{O}(1)$). This provides VSF the ability to calculate features with filters of any size at constant speed. Using this ability, VSF calculates the same features as VOCUS but without the need to down sampling the original image. Instead of keeping the same size of the filters and resizing the original image as VOCUS does, VSF uses filters of different size (emulating the ones from VOCUS) applied at the original image resolution. The overhead of VSF is the initial calculation of the integral image, however, this needs to be calculated only once by a straight-forward scan over the whole image.

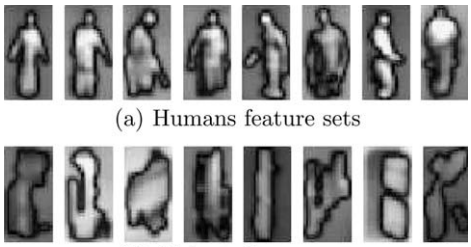
4.4. Object classification module

The system uses a feedforward neural network trained with backpropagation in order to classify objects. The inputs of this



(a) Original Image (b) VOCUS Off-Center Surround (c) Proposed Off-Center Surround

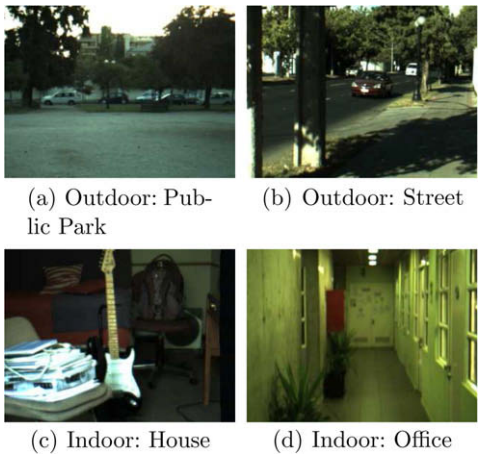
Fig. 7. Image details. VOCUS (center) only calculates coarse grained center–surround differences, loosing important details of the image. Instead, VSF (right) uses all the available information to provide a fine grained feature map.



(a) Humans feature sets

(b) Objects feature sets

Fig. 8. Examples of VSFs. Top row shows the resulting VSFs for several humans in different positions. Bottom row shows the resulting VSFs sets for random objects such as windows, walls, and doors.



(a) Outdoor: Public Park (b) Outdoor: Street (c) Indoor: House (d) Indoor: Office

Fig. 9. Different environments where the system was run.

module are the VSFs, scaled into 18×36 pixels. Example images can be seen in Fig. 8. The neural network is composed of three layers: (i) the input layer with 648 neurons, one for each pixel, (ii) the hidden layer with 5 neurons, and (iii) the output layer with 2 neurons, where the likelihood of being a human or non-human is stored. The criteria for human acceptance is calculated by comparing both output neurons. The one that has the largest value represents the output of the neural net. In order to obtain faster calculations Sigmoid activation functions were used. Therefore every pixel in the input image has to be scaled into the $\{-1 \ldots 1\}$ range. The values selected are similar to standard ones used in other works that rely on a neural network for object classification [13].

5. Implementation and results

In order to test the proposed system, we measure its performance on three main experiments: (i) detection of humans under different poses, (ii) comparison with a commonly used technique to detect humans based on face detection, and (iii) comparison with other human detection systems based on different visual features.

The system was mounted on top of a mobile robot. This robot was exposed to real life scenarios in different human inhabited environments. Fig. 9 shows some of the environments used to test our method. The base of the robot is a Pioneer P3-DX. A stereo camera is placed on top of the robot and it is able to take up to 30 frames per second. The stereo camera is configured in order to deliver valid range data from around 1.5–10 m. The system is expected to operate in non-crowded scenarios. The stereo camera is placed on top of a pan-tilt system. A saccadic control to manipulate the motion of the camera is used. The stereo camera is connected via firewire directly to a notebook which processes the information in addition to controlling the pan-tilt system. The system is implemented in C++ on a laptop with an AMD Sempron 1.79 GHz, 1.12 GB of RAM, running Linux Ubuntu 7.04 with a firewire interface to the stereo camera. This implementation allows us to achieve real time operation.

As training data, several images were acquired from diverse environments segmenting different objects from the scenes. In order to obtain these images, the system wandered around those environments and stored every segmented object. Using this output, the authors manually labeled 2239 object images, where 985 of them represented humans and 1254 represented non-human objects, such as trees, windows, doors, light posts, etc. Images were taken in both indoor and outdoor locations, such as offices, parks, and streets. In the set, humans appear under six different poses: frontal, back, and profile, with full and upper body views for each case. An example of these poses can be seen in Fig. 10 and further examples can be seen in Fig. 12. Tables 1 and 2 present further details of the training images. Example training images can be seen in Fig. 11. Each image has been masked with the available stereo information and resized to a common size of 18×36 pixels. The database is publicly available at http://samontab.googlepages.com/pedDatabase.rar.

It can be seen in Table 1 that the number of training images for frontal poses are slightly higher than the rest. This is explained because of the social nature of the robot. While obtaining training data, most people appeared in front of the robot staring at it. This difference in proportion that naturally appeared is kept in the

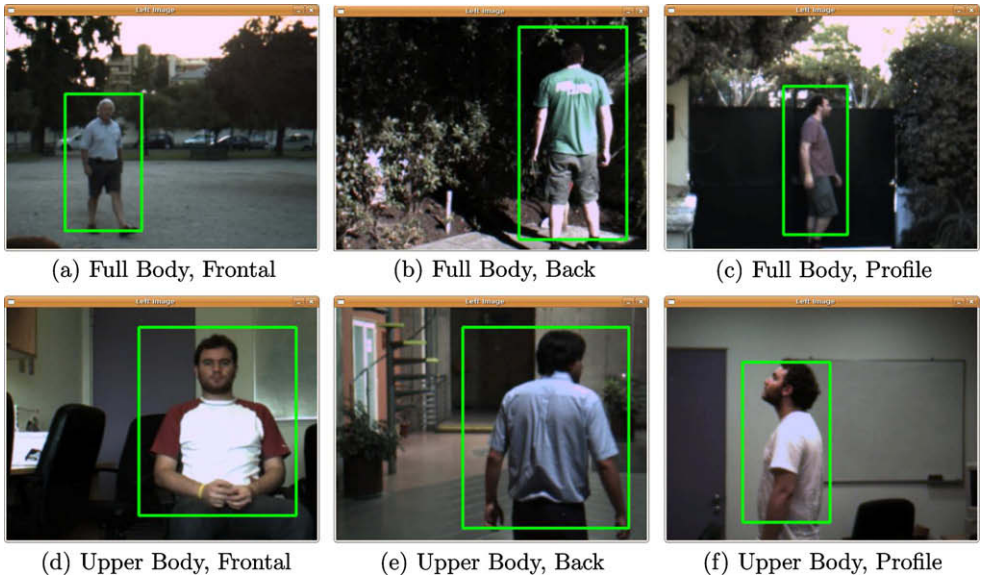


Fig. 10. Different poses detection. These images show the different poses that the system can detect.

Table 1
Poses of training images. This table shows the number of training images for each pose the system can detect.

	Frontal	Back	Profile
Full body	194	113	102
Upper body	351	137	88

Table 2
Training set details.

Environment	Type	Human examples	Non-human examples
Public park	Outdoor	148	314
Street	Outdoor	134	392
House	Indoor	246	117
Office	Indoor	457	431

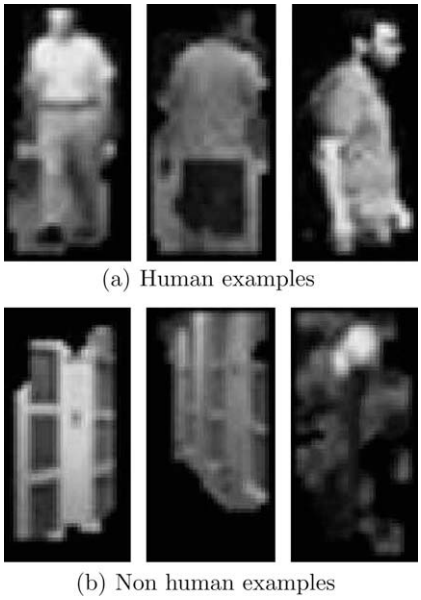


Fig. 11. Training set. Examples of the training images used in this work. Top row shows humans while bottom row shows non-human objects.

training set in order to represent the real type of interaction expected with people.

5.1. Different poses detection

The idea of this experiment is to test the performance of the system under different human poses. According to the training set described before, each human detection was archived in one of the following six categories: (i) full body, frontal; (ii) full body, back; (iii) full body, profile; (iv) upper body, frontal; (v) upper body, back, and (vi) upper body, profile. As this system is designed for a social robot with live video input instead of still images, every detection is considered only as one per human per pose. This means that, given a video sequence, a single human can only generate up to six detections in the system, one for each pose.

Tables 3 and 4 present detailed information about system performance in this experiment. It can be seen from these results that the system can detect humans regardless of their pose. As these results were obtained in different types of environment, the system can be used for both indoor and outdoor applications, although bad illumination can affect the performance of the system, mostly in the segmentation module, as stated in previous works with the same hardware [28].

5.2. Human face vs complete human body detection

In this experiment, the system was tested in order to compare the overall detection rate of a face detection system compared to that of the proposed system using live camera input in the same scenarios as the previous test: a public park, in the street, in a house, and in an office environment. People appear in the scene naturally, therefore different poses are presented. Each object detection is counted only once. Consecutive detections of the same object, whether it is a human or not, are not considered. The Viola Jones face detector (VJFD) system is selected for the test as it is commonly used in human detection for social robots [14,2,20]. We use the openCV implementation of VJFD [25]. Although this implementation was trained with a different image set, our goal here is to show how in common environments where social robots can operate, a general detector able to detect humans in different

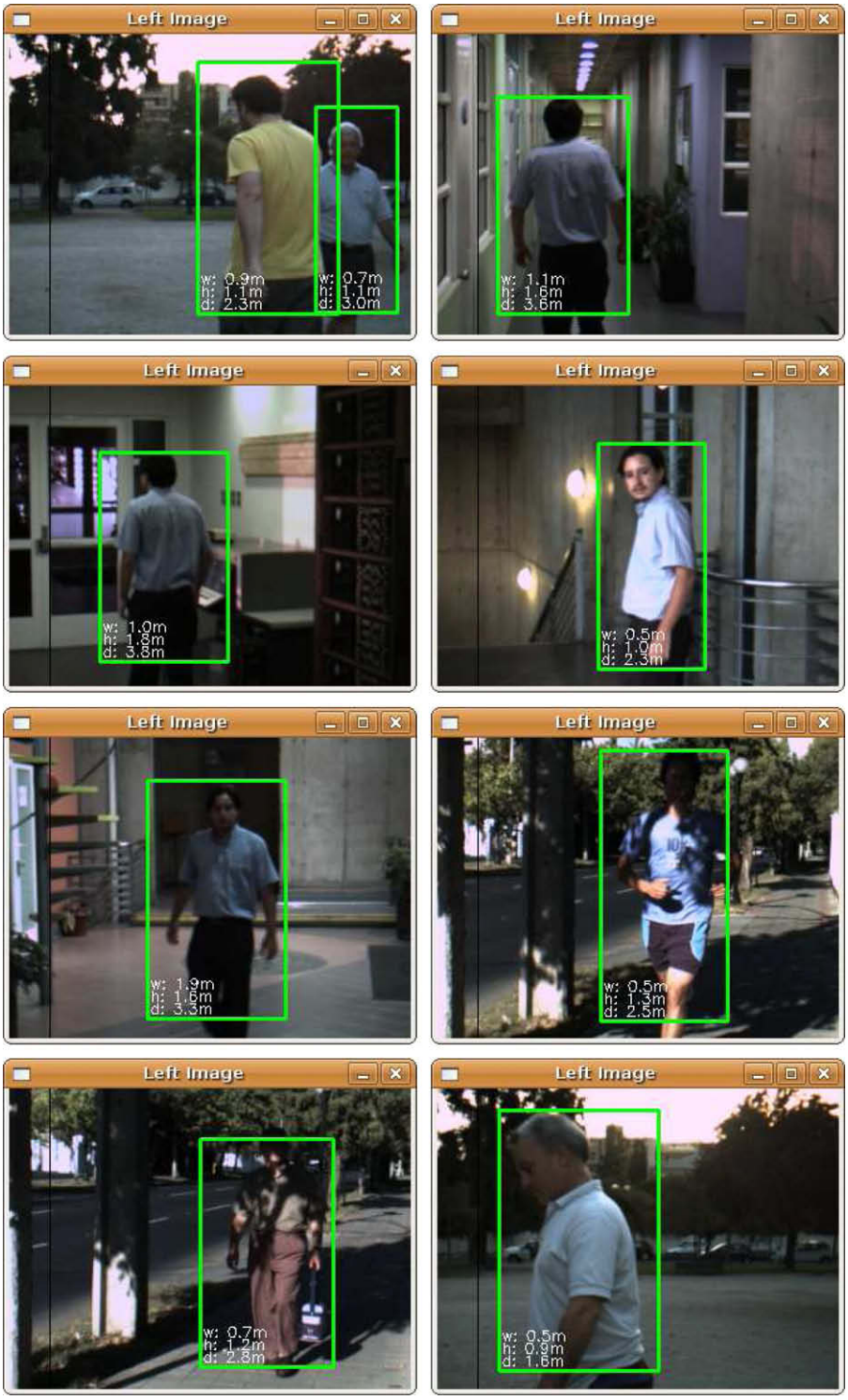


Fig. 12. Detection examples. The images show different detections of the system for human under different poses.

poses can be of great help. Detailed results of this experiment are presented in Table 5.

Table 6 shows that the proposed method provides higher detection rates and less false positives than the Viola Jones face detector (Figs. 13 and 14). In Table 7 can be seen that the largest difference in performance is obtained in outdoor environments. This can be explained because people is less restricted in outdoor environments, they can appear in many different poses and distances to the camera. As face detection systems are restricted only to detect humans facing the camera, their detection rate should drop if humans appear in different poses. Table 6 also shows that false positives are very low for the proposed method compared to Viola Jones. This is due in part to the fact that the proposed method uses real world measurements based on stereo vision and shape constraints in order to filter out non-human like regions.

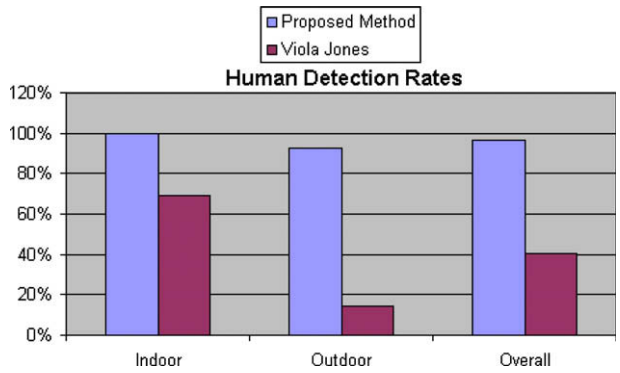


Fig. 13. Human detection rates. Comparison of the human detection rates between the proposed method and the Viola Jones face detector.

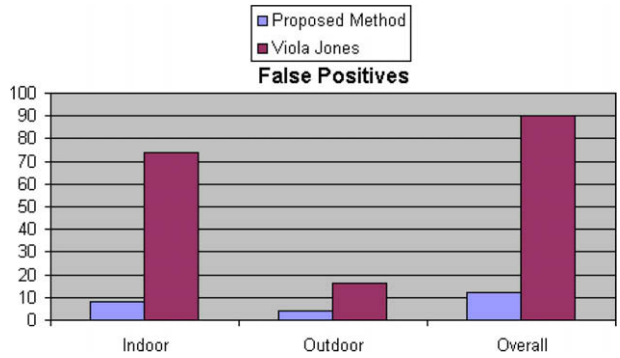


Fig. 14. False positives comparison. Comparison of the false positives between the proposed method and the Viola Jones face detector.

Table 3
Detailed performance of the proposed system for human detection in different poses: front, back and profile views of the front body and the upper body.

	Full Body Front	Full Body Back	Full Body Profile	Upper Body Front	Upper Body Back	Upper Body Profile
People detected	11	5	6	13	13	8
People missed	1	0	0	2	1	0

Table 4
Overall performance of the proposed system for human detection in different poses.

People-poses detected	Total	Det rate (%)	FP
56	60	93.3	8

Table 5
Comparison results. The table presents the result of a comparison between the proposed system and the Viola Jones face detection system.

System used	Environment	People	PD	FP
Proposed method	Public park	7	7	1
Proposed method	Street	7	6	3
Proposed method	House	4	4	6
Proposed method	Office	9	9	2
Viola Jones	Public park	7	0	4
Viola Jones	Street	7	2	12
Viola Jones	House	4	3	25
Viola Jones	Office	9	6	49

Table 5 shows that the proposed system only missed one person. This situation is presented in Fig. 15 where an almost full occlusion prevents the system from correctly detecting both persons.

Table 6
Overall comparison results. The table presents the general results of the comparison between the proposed system and the Viola Jones face detection system.

System used	Det rate (%)	FP
Proposed method	96.3	12
Viola Jones	40.7	90

Table 7
Comparison results. The table presents the result of a comparison between the proposed system and the Viola Jones face detection system in real world scenarios considering indoor and outdoor scenarios.

System used	Environment	Det rate (%)	FP
Proposed method	Indoor	100.00	8
Proposed method	Outdoor	92.86	4
Viola Jones	Indoor	69.23	74
Viola Jones	Outdoor	14.29	16

Fig. 15. Missed detection. An almost full occlusion prevents the system from correctly detecting both persons.

5.3. Feature comparison

In this experiment, we test the proposed VSFs against different visual features previously used for human detection. In particular, we choose to compare against features derived by: edge detectors, intensity gradients [40], and Haar-like wavelets [34]. Furthermore, we include as features the result of a traditional saliency map such as VOCUS. Edge detectors are one of the earliest features used for object detection. Although these features can represent the shape of an object, they are not robust to noise. [40] proposed the use of intensity gradient in order to obtain higher flexibility. In a related approach, recently [6] used histograms of oriented intensity gradients to detect humans. [34] made popular the use of Haar-like wavelets for object detection. The main drawback of these features is that filters size and shape are often user defined. In this paper we use the values used in [27]. As we detailed before, the proposed VSFs use predefined filters shape and size, based on biological findings about the operation of the human visual system.

Exactly the same detection test is performed for all the features under evaluation. K-fold cross-validation over the training set is used in order to measure the system classification error estimate. The number of folds used in this experiment is ten as suggested in [38]. Therefore, each test set is conformed of 223 random images and the other 2016 images are used as the training set.

Results of the comparison are presented in Table 8. Also Fig. 16 shows the resulting ROC curves. It can be seen that the proposed VSFs present a higher detection rate and a lower false positive rate than the previously used features. Also, it can be seen that traditional attention features such as VOCUS, perform very poorly due to the coarse grained feature maps they provide.

6. Conclusions and future work

In this paper, we propose and test a human detection system designed to operate onboard mobile platforms, such as a mobile robot. Using a stereo vision based segmentation algorithm, novel visual features based on a saliency mechanism (VSFs), and a neural network based classifier, the resulting system is able to: (i) detect humans on different poses, (ii) operate onboard a mobile platform, and (iii) operate in real time. After testing the system under real world conditions on several human inhabited environments, such as a public park and indoor buildings, our results indicate an average detection rate of 94.73% and a false positive rate of 6.15%, as estimated using 10-fold cross-validation.

Among the main novelties of the proposed approach is a center–surround saliency mechanism used to directly obtain the main visual features used for human detection. This type of saliency mechanism has been used before but as part of visual attention systems where the main goal is to detect interesting regions of an input image, therefore, feature map details are less relevant than timely computation. In our case, by using an efficient implementation of center–surround differences through the so-called integral image, we demonstrate a method to generate fine grained feature maps of visual saliency operating in real time at the original image resolution.

Table 8
Feature comparison. The table presents the comparison of results obtained using the default parameters for different features previously used for human detection against a traditional saliency map (VOCUS) and the proposed VSFs. Note that while a traditional saliency map such as VOCUS perform very poorly, VSFs present the best results.

Feature type	Det rate (%)	FP rate (%)
Edges	89.14	11.63
Intensity gradient	89.95	10.37
Haar like	92.39	7.26
VOCUS	79.84	34.08
VSFs	94.73	6.15

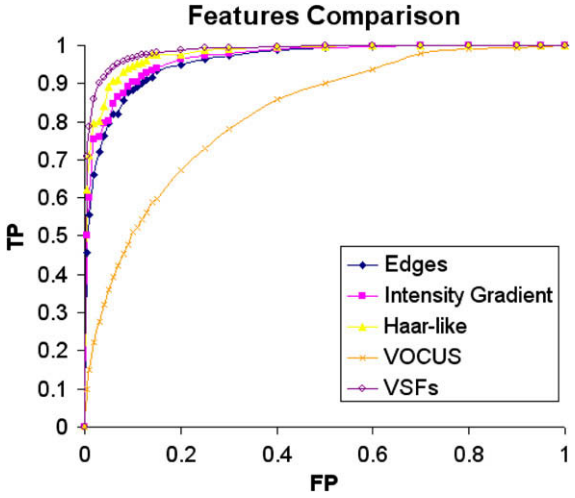


Fig. 16. ROC curves for human detection using different visual features.

One of the main robustness of the proposed approach is its capacity to detect humans under different poses. In particular, we test the advantage of such flexibility by contrasting the proposed system against a popular face detector algorithm that is commonly used in the mobile robotics arena to detect humans. Our results on common situations faced by a mobile robot show a highly significant increase in human detection rate. In this way, by using the proposed system, a robot can significantly increase its social capabilities by not requiring to detect just the humans that are facing it.

We also test the new proposed VSFs against previously visual features used for human detection, such as intensity gradient or the common Haar-like features. Our result indicates that VSFs outperforms previous features. In particular, we believe that VSFs have applications further beyond human detection, as relevant visual features for other visual classification problems.

As a future work, detection of humans in crowded scenarios should be considered. Also, the benefits of the use of a fine grained saliency map in other areas can be investigated.

Acknowledgments

This work was partially funded by FONDECYT grant 1070760 and Anillos ACT–32.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.imavis.2009.06.006.

References

[1] H. Asoh, Y. Motomura, F. Asano, I. Hara, S. Hayamizu, K. Itou, T. Kurita, T. Matsui, N. Vlassis, R. Bunschoten, B. Krose, Jijo-2: an office robot that communicates and learns, IEEE Intelligent Systems 16 (5) (2001) 46–55.
[2] N. Bellotto, H. Hu, Multisensor integration for human–robot interaction, IEEE Journal of Intelligent Cybernetic Systems 1 (2005).
[3] M. Bennewitz, F. Faber, D. Joho, M. Schreiber, S. Behnke, Towards a humanoid museum guide robot that interacts with multiple persons, in: 5th IEEE-RAS International Conference on Humanoid Robots, 2005, pp. 418–423.
[4] W. Burgard, A. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, Experiences with an interactive museum tour-guide robot, Artificial Intelligence 114 (1999) 1–2.
[5] F. Crow, Summed-area tables for texture mapping, Proceedings of SIGGRAPH 18 (3) (1984) 207–212.
[6] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-05), 2005, pp. 886–893.
[7] P. Espinace, D. Langdon, A. Soto, Unsupervised identification of useful visual landmarks using multiple segmentations and top-down feedback, Robotics and Autonomous Systems 56 (6) (2008) 538–548.
[8] T. Fong, I. Nourbakhsh, K. Dautenhahn, A survey of socially interactive robots, Robotics and Autonomous Systems 42 (3-4) (2003) 143–166.
[9] S. Frintrop, VOCUS: A Visual Attention System for Object Detection and Goal-directed Search, PhD thesis, Rheinische Friedrich-Wilhelms-Universitat Bonn Germany, 2005.
[10] S. Frintrop, M. Klodt, E. Rome, A real-time visual attention system using integral images, in: Proceedings of the 5th International Conference on Computer Vision Systems, 2007.
[11] T. Gandhi, M.M. Trivedi, Pedestrian protection systems: issues, survey, and challenges, IEEE Transactions on Intelligent Transportation Systems 8 (3) (2007) 413–430.
[12] D.M. Gavrila, J. Giebel, S. Munder, Vision-based pedestrian detection: the protector system, in: Proceedings of the IEEE Intelligent Vehicle Symposium, 2004, pp. 13–18.
[13] D. Geronimo, A. Lopez, A. Sappa, Computer vision approaches to pedestrian detection: visible spectrum survey, Lecture Notes In Computer Science 4477 (2007) 547–554.
[14] G. Hollinger, Y. Georgiev, A. Manfredi, B. Maxwell, Z. Pezzementi, B. Mitchell, Design of a social mobile robot using emotion-based decision mechanisms, in: International Conference on Intelligent Robots and Systems, 2006.
[15] Y. Huang, S. Fu, C. Thompson, Stereovision-based object segmentation for automotive applications, EURASIP Journal on Applied Signal Processing 14 (2005) 2322–2329.

[16] L. Itti, The ilab neuromorphic vision C++ toolkit: free tools for the next generation of vision algorithms, The Neuromorphic Engineer 1 (1) (2004) 10.
[17] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Pattern Analysis and Machine Intelligence 20 (11) (1998) 1254–1259.
[18] T. Kadir, A. Zisserman, M. Brady, An affine invariant salient region detector, in: Proceedings of the European Conference on Computer Vision (ECCV-04), 2004.
[19] K. Konolige, The SRI small vision system. Available from: <http://www.ai.sri.com/~konolige/svs/>.
[20] S. Lang, M. Kleinehagenbrock, S. Hohenner, J. Fritsch, G.A. Fink, G. Sagerer, Providing the basis for human–robot-interaction: a multi-modal attention system for a mobile robot, in: International Conference on Multimodal Interfaces, 2003, pp. 28–35.
[21] M.J. Mataric, J. Eriksson, D.J. Feil-Seifer, C.J. Winstein, Socially assistive robotics for post-stroke rehabilitation, Journal of Neuroengineering and Rehabilitation 4 (5) (2007).
[22] K. Mikolajczyk, C. Schmid, An affine invariant interest point detector, in: Proceedings of the European Conference on Computer Vision (ECCV-02), 2002.
[23] R. Munoz-Salinas, E. Aguirre, M. Garcia-Silvente, People detection and tracking using stereo vision and color, Image and Vision Computing 25 (6) (2007) 995–1007.
[24] N. Ogale, A Survey of Techniques for Human Detection from Video, Master's thesis, University of Maryland, 2006.
[25] Intel Corporation. OpenCV: Open Source Computer Vision Library. Available from: <http://www.intel.com/research/mrl/research/opencv/>.
[26] S.E. Palmer, Vision Science: Photons to Phenomenology, MIT Press, 1999.
[27] C. Papageorgiou, T. Poggio, A trainable system for object detection, International Journal of Computer Vision 38 (1) (2000) 15–33.
[28] S. Pszczolkowski, A. Soto, Human detection in indoor environments using multiple visual cues and a mobile robot, Lecture Notes in Computer Science, Springer, 2007.
[29] A.N. Rajagopalan, P. Burlina, R. Chellappa, Detection of people in images, in: Proceedinds of the IEEE International Joint Conference on Neural Networks (IJCNN'99), Washington, 1999.
[30] A.N. Rajagopalan, R. Chellappa, Higher order statistics-based detection of people/vehicles in images, INSA-A Journal: Special Issue on Image Processing, Vision and Pattern Recognition 67 (2001) 157–166.
[31] F. Schaffalitzky, A. Zisserman, Multi-view matching for unordered image sets, or how do I organize my holiday snaps?, in: Proceedings of the European Conference on Computer Vision (ECCV-02), 2002.
[32] A.M. Treisman, G.A. Gelade, Feature-integration theory of attention, Cognitive Psychology 12 (1980) 97–136.
[33] V. Vezhnevets, V. Sazonov, A. Andreeva, A survey on pixel-based skin color detection techniques, in: Proceedings of Graphicon-03, 2003, pp. 85–92.
[34] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR-01), 2001, pp. 228–235.
[35] P. Viola, M. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, in: International Conference on Computer Vision (ICCV03), 2003.
[36] D. Walther, U. Rutishauser, C. Koch, P. Perona, On the usefulness of attention for object recognition, in: Proceedings of the European Conference on Computer Vision (ECCV-04), 2004.
[37] T. Wilhelm, H.J. Bohme, H.M. Gross, A multi-modal system for tracking and analyzing faces on a mobile robot, Robotics and Autonomous Systems 48 (1) (2004) 31–40.
[38] I. Witten, F. Eibe, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann Publishers, 2000.
[39] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.
[40] L. Zhao, C.E. Thorpe, Stereo and neural network-based pedestrian detection, IEEE Transactions on Intelligent Transportation Systems 1 (3) (2000) 148–154.