International Journal of Humanoid Robotics

Dedicated Mechanisms for the Attention System in Humanoid Robots

Markus S. Schlosser and Kristian Kroschel
Institut für Nachrichtentechnik
Universität Karlsruhe
76128 Karlsruhe, GERMANY
{schlosser, kroschel}@int.uni-karlsruhe.de

A humanoid robot must be able to cooperate efficiently and safely with humans in an unconstrained environment. Furthermore, it should not only present no danger to humans itself, but also be able to detect dangerous situations for the user. To restrict the processing only to salient objects and events, attention systems modeling the human visual system have been proposed. However, they are computationally demanding and the capabilities of digital signal processing systems are still rather limited compared with those of humans. Therefore, we propose that simple dedicated mechanisms should be introduced to identify important or even dangerous events. To this end, efficient detectors for objects falling down, objects moving at high velocity as well as human faces are presented in this work.

Keywords: Attention system; alerting system; humanoid robot; safe cooperation; danger detection; dedicated mechanisms.

1. Introduction

It is computational prohibitive for a robot to process every region in every captured image to the highest cognitive levels, like object recognition and action planning. It has to bundle its limited resources on regions likely to contain important objects or events. The objective of an attention system is to detect these salient objects and events and, thus, to allow to learn efficiently about a new environment and to react to important incidents.

As the human brain faces the same problem and performs remarkably well on this task, it seems sensible to use it as a starting point. The neural processes in the human visual system are not completely understood yet, but psychological studies suggest that objects are selected based on an interplay of bottom-up, image-based saliency cues and top-down, task-dependent cues.[1,2] The bottom-up part is sufficiently well understood that computer models simulating its behavior have been proposed, e.g. by Cave[3], Itti[4] and Wolfe et al.[5] However, not very much is known about the top-down part. It is typically implemented as an adjustment of the weights in the bottom-up system.

In the bottom-up part of the human visual system several topographical feature maps like intensity, opponent-color, orientation, depth and movement are extracted

in parallel over the complete visual field. These maps are combined into a single saliency map. Regions are thereby regarded as salient if the center differs markedly from its surroundings.

To this end, the maps pass through a hierarchical structure with higher levels consisting of smaller maps and consistently larger receptive fields per unit. On the one hand, this hierarchy calculates the center-surround differences and performs the fusion into a single saliency map. On the other hand, the hierarchy also provides for scale invariance. The region with the highest activation in the saliency map is then chosen as the next point to focus attention on.

The locations predicted by these models have been shown to be a good approximation of the ones attended by human subjects during visual search tasks. However, they are quite complex to compute and the capabilities of digital signal processing systems are still rather limited compared with those of humans. This is especially true if the system needs to be mobile, like e.g. a humanoid robot.

It is imperative for a humanoid robot not only to cooperate efficiently but also safely with humans in an unconstrained environment. Furthermore, it should not only present no danger to humans itself but also be able to detect dangerous situations. It is rather unlikely that this can be achieved with the aforementioned attention systems in the near future.

It is reasonable to use these systems in situations that are not time-critical, e.g. to explore a new environment efficiently. However, in our opinion, they should be augmented by simple dedicated mechanisms to identify dangerous situations. A similar idea was already presented by Milanese et al.[6], where an alerting system performing motion detection was introduced to supplement a bottom-up attention system based on color, orientation and local curvature. On the other hand, this alerting system detected any movements and not only dangerous ones.

In this work efficient detectors for several important or dangerous situations are presented. Sect. 2 deals with the identification of objects falling down. In Sect. 3 a detector for objects moving at high velocity and in Sect. 4 a detector for human faces are presented.

Our system consists of a color camera having a field of view of approximately 55 degrees that captures images of 320x240 pixels in size at a frame rate of 30 images per second. The camera is attached to a Pentium IV with 1.8 GHz.

2. Falling Object Detection

An object falling down is a dangerous or, at least, an exceptional event that deserves closer attention. The user could have dropped something or he could have thrown something on the floor without having noticed it. Furthermore, the robot could have dropped something itself, too. Therefore, a simple but effective detector for falling objects based on difference images is presented in this section. Fig. 1 shows a typical result, where a person drops a mug while moving to the left.

Fig. 2 presents the different processing steps in detecting this falling object. First,

(a) Detected falling object (b) Previous image

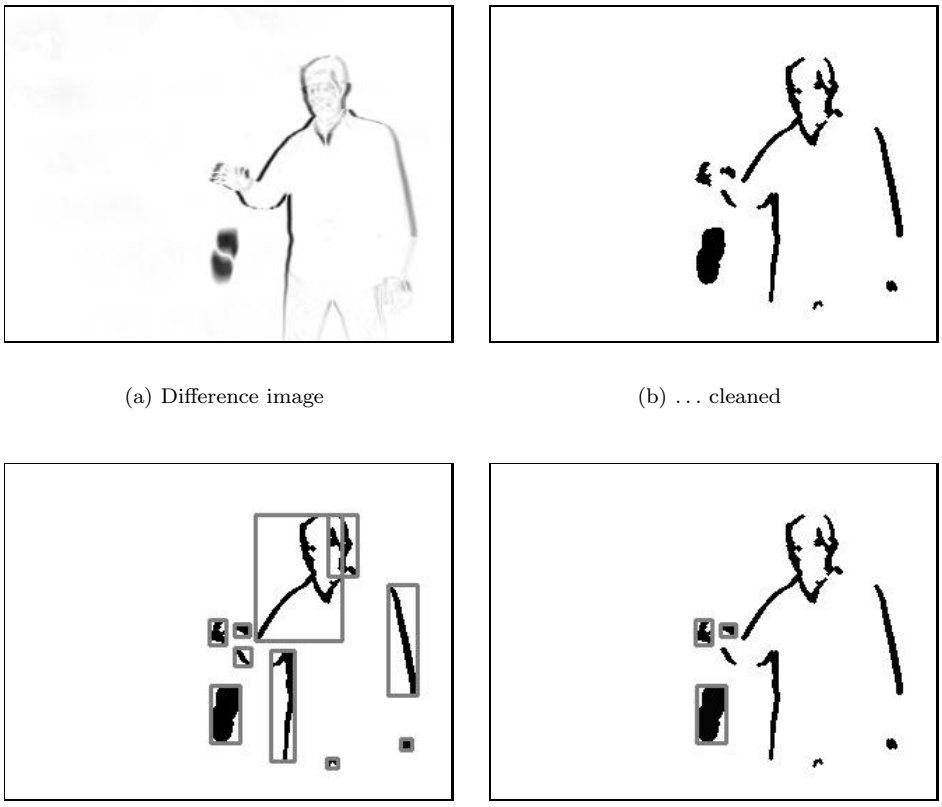
Fig. 1. Detection of an object falling vertically

a difference image between the current and previous frame is calculated. For higher robustness, this is done on each of the three channels of the color image and the results are combined into one gray scale image of 8 bit depth (see Fig. 2(a)). Second, this difference image is thresholded and morphological operations are performed to clean it up. The resulting black and white image is shown in Fig. 2(b). A closing operation is used to fill in gaps within one object, as can be seen e.g. for the mug and for the face. Thereafter an opening operation is used to remove very small structures which would complicate the subsequent extraction of the outer contours of the black blobs unnecessarily.

The next step consists of replacing the outer contours by their bounding rectangles to facilitate the further treatment (see Fig. 2(c)). Finally, small objects as well as bounding rectangles containing only few black pixels are removed, as can be seen in Fig. 2(d). Additionally, rectangles being similar in size and below each other are merged into a single one as falling objects are likely to produce to distinct non-zero regions. This can either be due to the object being sufficiently fast to occupy a completely distinct region in the next image or be due to a falling single-colored object whose positions in two consecutive images are still overlapping, as can be seen in Fig. 2(a). The bounding rectangle indicating the falling object in Fig. 1(a) is produced by restricting the height of the rectangle in Fig. 2(d) to the bottom line of the previous rectangle.

To decide if a falling object is present or not, the rectangles obtained in one frame are compared with those of the previous frame. A falling object represents an accelerated, downward movement. In terms of the bounding rectangles, this is equivalent to

- a downward vertical displacement,
- an increase in height,



(a) Difference image (b) ... cleaned

(c) Bounding rectangles (d) ... cleaned

Fig. 2. Processing steps in falling object detection

- a similar horizontal position,
- and a similar width.

The first condition is self-evident. The second one takes into account the increase in height due to the increase in velocity of a falling object over time. The higher velocity not only leads to a steady increase in distance between the previous and current position of the object but also to a bigger height of the object itself due to the finite exposure time for capturing the images.

The last two conditions require the horizontal position and the width only to be similar to allow for inaccuracies in the process and to detect objects that are rotating or not falling perfectly vertically, too. A more severe example for this is shown in Fig. 3.

To distinguish between an object falling down and an object being actively

(a) Detected falling object (b) Difference image

Fig. 3. Detection of an object falling in an inclined way

moved downwards, it was first tried to verify the law of free fall in the vertical displacement and the increase in height. However, this did not show a robust performance. Especially, parts of the hand being included in the bounding rectangle of the falling object as well as rotations of the object causing changes in its height resulted in severe problems in determining its acceleration. Furthermore, an additional image was needed to estimate the acceleration requiring a longer observation interval.

On the other hand, an active downward movement typically does not fulfill the conditions mentioned above. As shown in Fig. 4, the rotational movement of the arm around a fixed point results either in an increasing vertical position but a decreasing height or an increasing height but a fixed vertical position. Furthermore, at the beginning and the end of the sequence, the bounding rectangles are discarded during the next processing step due to an insufficient amount of moved pixels. Therefore, false detections are limited to cases where the arm is moved almost exactly in the direction of the camera. This was regarded as acceptable as such a movement would have been difficult to classify correctly with the more complex approach of verifying the law of free fall, too.

The thresholds were adjusted so that the mug shown in the figures was robustly detected to be falling for distances ranging from 0.7 m to 2.5 m. For larger distances the mug becomes so small that it is removed from the images during the cleanup. For distances smaller than 0.7 m, the mug is typically not visible long enough without interferences of the moving hand. As far as the maximum allowed inclination is concerned, Fig. 3 presents the worst case. In similar sequences the object is not always detected as falling.

The determination of the thresholds is not very critical. They can be changed significantly and still a large variety of falling objects are robustly detected for a

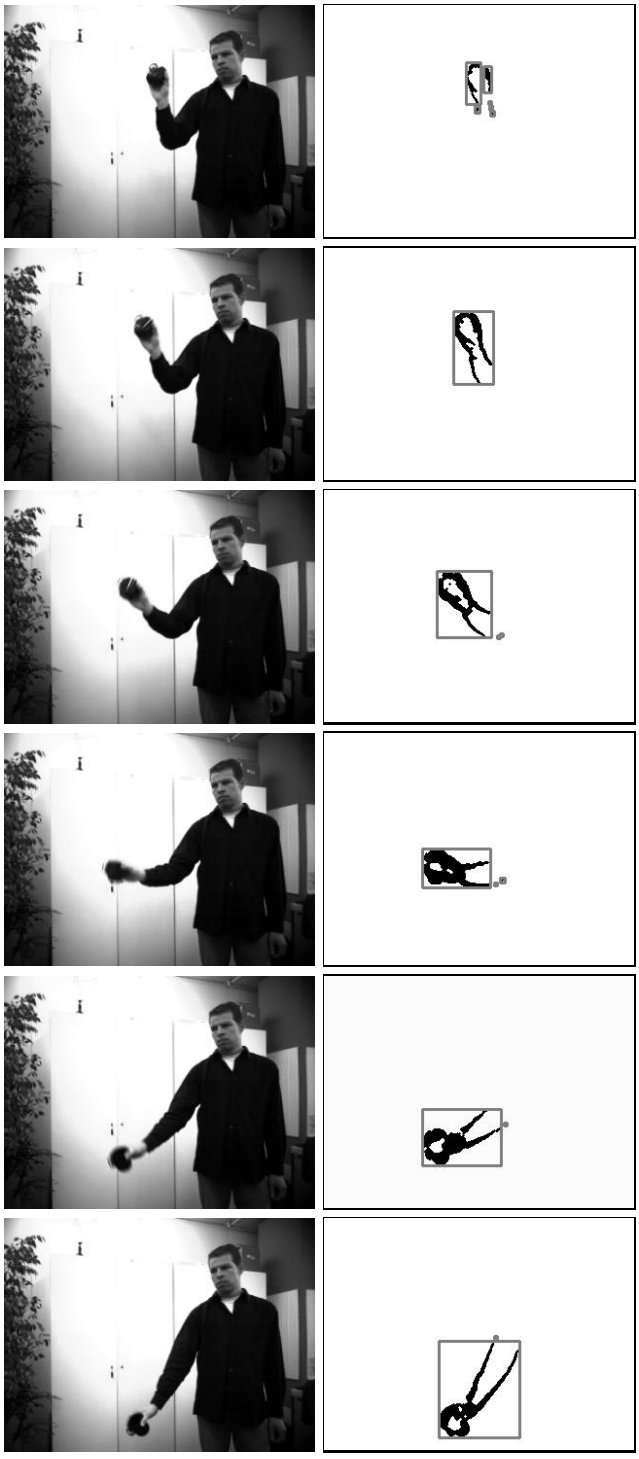


Fig. 4. Image sequence of an object being actively moved downwards (left column: current input image; right column: difference image with bounding boxes before clean up and merging)

wide interval of ranges. More severe thresholds result in a lower false alarm rate but also in the objects being needed to fall more vertically as well as changes in their shape caused by inaccuracies and rotations being less acceptable. With our settings, the false alarm rate is approximately one misclassification in 250 images for a typical sequence of a person moving in front of the camera.

The algorithm is very efficient. It runs at approximately 50 frames per second on our system.

3. High Velocity Detection

Objects moving at high velocity not only represent impending danger themselves naturally but a quick movement of the user may also indicate a dangerous situation, which the robot has missed to realize itself. To this end, a simple algorithm to detect such events based on optical flow images is presented in the following (see Fig. 5). A different approach compared with Sect. 2 was used as nothing is known about the direction of the movement.

(a) Input image (b) Detected object

Fig. 5. Detection of an object moving at high velocity

Optical flow is calculated by comparing the change in brightness between two consecutive frames at one point with the brightness gradient at that point. Therefore, it can primarily be calculated at a non-zero brightness gradient, parallel to this gradient and only for displacements smaller than its extent. Velocities in other directions and at other pixels can only be estimated under the further assumption of smoothness of the motion field.[7]

Objects moving at high velocity result in large displacements from one image to the next and, thus, they are difficult to handle with optical flow algorithms. To be able to calculate an accurate optical flow field nevertheless, we use the multi-scale approach described in Bergen et al.[8] The optical flow is first computed at a

low resolution. Thereafter, the input image is warped according to the calculated motion field and the optical flow is refined at the next higher resolution. Then the input image is warped again, and so on. In our application, a Laplacian pyramid consisting of three levels is used.

(a) Downsampled input image (b) Optical flow image

Fig. 6. Processing steps in high velocity detection

As no exact motion field is required for our application and as the computational expense of calculating an optical flow image is directly proportional to the size of the image, the input images are low-pass filtered and downsampled twice yielding a reduction in image size and, accordingly, computation time by a factor of 16. Fig. 6(a) displays the thus obtained image of 80x60 pixels corresponding to Fig. 5(a).

Based on the calculated optical flow field (see Fig. 6(b)), the decision if a fast moving object is present or not seems straightforward. However, an optical flow field never represents a one to one mapping of the velocity distribution in an image. In our case this is aggravated by the reduction in the resolution of the input images. The velocities, which are assigned to pixels belonging to a moving object, are rather broadly distributed and, thus, it is difficult to estimate the proper velocity of an object. As a result, the distribution of velocities over an object as a whole is examined to check if a large percentage of pixels are assigned a high velocity.

Additionally, already the definition of a fast moving object is difficult. The definition is not only subjective but also depends on the size of an object. Clearly, a smaller object needs to move at a higher speed to be called “fast moving”. Therefore, several decision rules were implemented to cope with the varying size of objects.

To detect large, fast moving objects covering most part of the image, like e.g. a person, the velocity distribution over the complete image is used. Pixels surpassing a low velocity threshold are thereby used as an indicator for the size of the object.

To detect smaller, fast moving objects, subwindows of 40x30 pixels and 30x20

pixels in size are slid over the optical flow image with approximately 50 % overlap in width and height to remove the effects of other moving objects being present in the scene. Pixels surpassing a low velocity threshold are again used to indicate the size of an object. As indicated above, a smaller size of the object thereby leads to a higher velocity threshold that is needed to be surpassed.

The thresholds were adjusted by several experiments so that good results were achieved for different objects, like e.g. a moving person, a waved hand, a hand waving an object, for distances ranging from 0.8 m to 2 m. A moving person and a waved hand are detected as fast moving if their velocities are higher than approximately 150 pixels per second and 300 pixels per second, respectively.

A common drawback of determining the velocity only in image coordinates, as e.g. done by our optical flow algorithm, is that the calculated velocity of an object not only depends on its velocity but also on its distance to the camera. The same movement of an object being performed closer to the camera results in a higher velocity in image coordinates. As the object will also become bigger, it will be detected more easily as moving at a high velocity. However, this is not problematic here as humans will also tend to be attracted more easily by a movement closer to the camera.

Another problem lies in the assumption underlying optical flow calculations that objects move on straight lines. Therefore, the rotational movement of a waved hand is somewhat problematic and the fingertips are sometimes not detected by the algorithm.

The algorithm is computationally very efficient again. On our system, it runs at a frame rate of approximately 30 frames per second.

4. Face Detection

Psychological studies based on the visual search paradigm, where human subjects are asked to search for an object among distractors in a briefly presented display, have shown that searches for faces are inefficient.[9] Therefore, faces are not among the basic features in the human visual system that are extracted pre-attentively over the whole field of view in parallel. On the other hand, it was equally shown that dedicated mechanisms to detect faces exist in the human brain.

Furthermore, humans and especially the user play an outstanding role for the behavior of a humanoid robot. Not only shall the robot present no danger to humans sharing its environment but it is also the robot's main purpose to serve humans. Finally, the detection of the user's face is also the first step in determining what he pays attention to and, thus, in deciding if the user has probably missed to realize a dangerous situation. Therefore, a special mechanism to detect human faces based on a combination of color information and a Haar-like feature based classifier is presented in this section.

The very fast object detector based on a cascade of simple classifiers using Haar-like features was introduced in Viola et al.[10], applied to face detection and its high

(a) Detected face in input image (b) Skin color image

Fig. 7. Detection of human faces

performance presented. Using a cascade has the advantage that many subwindows can be classified as not containing a face and thus discarded early in the processing chain. Only subwindows containing faces or something very similar to faces need to pass through the complete cascade to be classified. This reduces the computation time drastically compared with a single but complex classifier. The required computation time was further reduced by using the Haar-like features. These simple features can be computed very efficiently using an integral image[10] but still provide a rich representation of the image contents.

Lienhart et al.[11] refined this approach by introducing rotated features and a post optimization procedure. This reduced the false alarm rate by about 24 % at a given hit rate. At a hit rate of 95 %, the false alarm rate was stated to be below 0.5 %. These results could be verified by tests using the FERET database[12]. Even faces being rotated by up to 25° are typically detected.

This detector running at approximately 5 frames per second on our system is at the heart of our special mechanism. It is sensible to combine it with a detector for skin color as the Haar-like features do not make use of the color information contained in the image.

It has been shown that human skin colors cluster in a small region in the color space and that they differ more in intensity than in color.[13] Skin color can be modeled as being normally distributed with a low variance in the chromatic color space and therefore most of the false alarms caused by non-skin-colored objects can be suppressed efficiently. The chromatic color space is obtained from the RGB color space by simple intensity normalization.

Furthermore, skin color information can be used to reduce the search region. In the example shown in Fig. 7, 80% of the subwindows are dropped even for the conservative threshold that at least 50% of the pixels contained in a subwindow need

to be skin-colored. Unfortunately, this classification based on skin color does not only require the pixels to be classified as skin-colored or not but also the amount of skin color to be determined for each frame. Although the latter can be done efficiently using the integral image, no significant improvements in processing time could be achieved.

Nevertheless, the introduction of skin color makes sense in our scenario. Our humanoid robot performs user tracking with the help of skin color, too,[14] so that the skin color classification has to be performed anyway. Additionally, the skin color image is processed further to form blobs. Therefore, it is planned to crop subimages around these blobs in the future and to preform face detection only on those subimages so that the calculation of the amount of skin color for each subwindow will be avoided as well.

5. Conclusions

If robots are to act in the same environment as humans, it is of utmost importance that they do not present any danger for the human users. Furthermore, they should be able to detect important or even dangerous situations. In our opinion it is rather unlikely that this can be achieved with the traditional attention systems in the near future. Therefore, three dedicated mechanisms have been introduced in this work to supplement these systems: efficient detectors for objects falling down, objects moving at high velocity as well as human faces. These detectors were shown to be robust and computationally inexpensive.

Future work will be aimed at integrating these mechanisms into our attention system based on the work of Itti[4] and into our face tracker[14]. Additionally, further dedicated mechanisms to detect dangerous situations are envisioned, e.g. tracking the hands of the user shall be used to detect exceptional movements and to predict interactions of the human with the environment.

Acknowledgments

This work is part of the Sonderforschungsbereich (SFB) No. 588 *"Humanoide Roboter - Lernende und kooperierende multimodale Roboter"* at the University of Karlsruhe. The SFB is supported by the Deutsche Forschungsgemeinschaft (DFG). Portions of the research in this paper use the Color FERET database of facial images collected under the FERET program.

References

1. L. Itti and C. Koch, Computational modeling of visual attention, *Nature Reviews Neuroscience* **2**(3), 194–203 (2001).
2. H. Pashler (ed.), *Attention* (London: UCL Press, 1996).
3. K. R. Cave, The featuregate model of visual selection. *Psychological Research* **62**, 182–194 (1999).

4. L. Itti, *Models of Bottom-Up and Top-Down Visual Attention* (PhD thesis, California Institute of Technology, 2000).
5. J. M. Wolfe and G. Gancarz, *Basic and Clinical Applications of Vision Science* (Dordrecht, Netherlands: Kluwer Academic, 1996), chapter Guided Search 3.0, pp. 189–192.
6. R. Milanese, H. Wechsler, S. Gill, J.-M. Bost, and T. Pun, Integration of bottom-up and top-down cues for visual attention using non-linear relaxation, in *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1994), pp. 781–785.
7. B. K. P. Horn and B. G. Schunck, *Determining optical flow* (A. I. Memo 572, Massachusetts Institute of Technology, 1980).
8. J. R. Bergen and R. Hingorani, *Hierarchical motion-based frame rate conversion* (Technical report, David Sarnoff Research Center, 1990).
9. J. M. Wolfe, *Attention* (London: UCL Press, 1996), chapter Visual Search, pp. 13–73.
10. P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001), pp. I-511 – I-518.
11. R. Lienhart and J. Maydt, An extended set of haar-like features for rapid object detection, in *Int. Conf. on Image Processing* (2002), pp. I-900 – I-903.
12. P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, The FERET evaluation methodology for face recognition algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 1090–1104 (2000).
13. J. Yang and A. Waibel, A real-time face tracker, in *3rd IEEE Workshop on Applications of Computer Vision* (1996), pp. 142–147.
14. K. Nickel and R. Stiefelhagen, Detection and tracking of 3D-pointing gestures for human-robot-interaction, in *3rd IEEE Int. Conf. on Humanoid Robots - Humanoids* (2003).