We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

22

Visual Attention and Distributed Processing of Visual Information for the Control of Humanoid Robots

Aleš Ude[1,2], Jan Moren[1], and Gordon Cheng[1,3]
[1]*ATR Computational Neuroscience Laboratories, Kyoto,* [2]*Jožef Stefan Institute, Ljubljana,*
[3]*Japan Science and Technology Agency, Saitama*
[1, 3]*Japan,* [2]*Slovenia*

1. Introduction

The function of visual attention is to identify interesting areas in the visual scene so that limited computational resources of a human or an artificial machine can be dedicated to the processing of regions with potentially interesting objects. Already early computational models of visual attention (Koch and Ullmann, 1985) suggested that attention consists of two functionally independent stages:

- in the preattentive stage features are processed rapidly and in parallel over the entire visual field until the focus of attention has been identified, which triggers the eye movement towards the target area;
- in the second phase, the computational resources are dedicated towards the processing of information in the identified area while ignoring the irrelevant or distracting percepts.

Visual attention selectivity can be either overt to drive and guide eye movements or covert, internally shifting the focus of attention from one image region to another without eye movements (Sun and Fisher, 2003). Here we are interested in visual attention that involves eye movements and how to implement it on a humanoid robot. Overt shifts of attention from one selected area to another were demonstrated for example in face recognition experiments (Yarbus, 1967). Although the subjects perceived faces as a whole in these experiments, their eye movements showed that their attention was shifted from one point to another while processing a face. The analysis of fixation points revealed that the subjects performed saccadic eye movements, which are very fast ballistic movements, to acquire data from the most informative areas of the image. Since high velocities disrupt vision and also because the signal that the target had been reached would arrive long after the movement had overshot, saccadic eye movements are not visually guided. The input to the motor system is the desired eye position, which is continuously compared to an efference copy of the internal representation of the eye position.

Many computational models of preattentive processing have been influenced by the feature integration theory (Treisman and Gelade, 1980), which resulted in several technical implementations, e. g. (Itti et al., 1998), including some implementations on humanoid robots (Driscoll et al., 1998; Breazeal and Scasselatti, 1999; Stasse et al., 2000; Vijayakumar et

Source: Humanoid Robots: Human-like Machines, Book edited by: Matthias Hackel
ISBN 978-3-902613-07-3, pp. 642, Itech, Vienna, Austria, June 2007

al., 2001). With the exception of (Driscoll et al., 1998), these implementations are mainly concerned with bottom-up, data-driven processing directed towards the generation of saliency maps. However, many theories of visual search, e. g. guided search, suggests that there are several ways for preattentive processing to guide the deployment of attention (Wolfe, 2003). Besides the bottom-up pointers towards salient regions, there is also a top-down guidance based on the needs of the searcher.

Figure 1. Humanoid head used in the experiments. It has a foveated vision system and 7 degrees of freedom (two DOFs in each eye and three DOFs in the neck)

Although bottom-up attention has been studied extensively in the past and is relatively well understood, it is still not easy to implement it in real-time on a technical system if many different feature maps are to be computed as suggested by the feature integration theory. One possible solution is to apply distributed processing to realize the extraction and analysis of feature maps in real-time. We have therefore developed a suitable computer architecture to support parallel, real-time implementation of visual attention on a humanoid robot. The guiding principle for the design of our distributed processing architecture was the existence of separate visual areas in the brain, each specialized for the processing of a particular aspect of a visual scene (Sekuler and Blake, 2002). It is evident from various visual disabilities that the ability of the brain to reassign the processing of visual information to new brain areas is rather limited and that it also takes time. Instead, visual information is transferred along a number of pathways, e. g. magnocellular pathway, parvocellular-blob pathway, and parvocellular-interblob pathway (Rolls and Deco, 2003), and visual processes are executed in well defined areas of the brain. Visual perception results from interconnections between these partly separate and functionally specialized systems. Thus our goal was to design a system that will allow us to transfer information from the source to a number of computers executing specialized vision processes, either sequentially or in parallel, and to provide means to integrate information from various streams coming at different frame rates and with different latencies. The transfer of information in the system can be both feed-forward (bottom-up processing) and feed-backward (top-down effects).

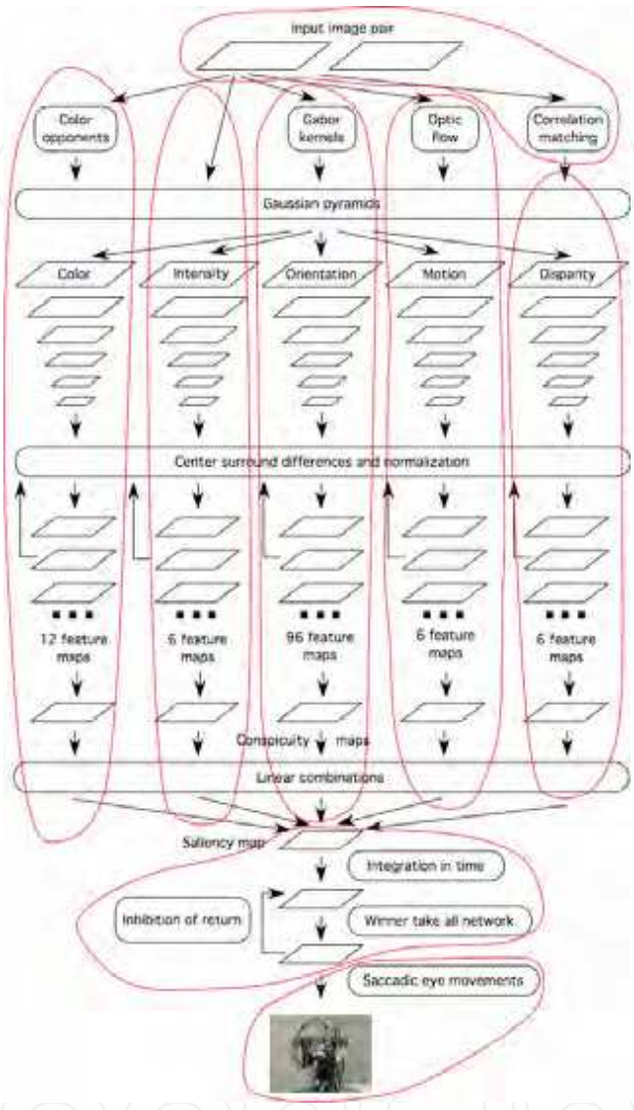


Figure 2. Bottom-up visual attention architecture based on feature integration theory. Compared to the architecture proposed by (Itti et al., 1998), there are two additional streams: motion and disparity. They are both associated with the magnocellular processing pathway in the brain, whereas color, intensity, and orientation are transferred along the parvocellular pathway. Red circles indicate the distribution of visual processes across the computer cluster. Each circle encloses the processes executed by one computer. Our system also includes the control of eye movements

Besides the distributed implementation to realize real-time behavior of the system for the control of a humanoid robot, we also studied the incorporation of top-down information into the bottom-up attention system. Top-down signals can bias the search process towards

objects with particular properties, thus enabling the system to find such objects more quickly.

2. Bottom-up preattentive processing

Figure 2 shows our distributed implementation of bottom-up visual attention, which is a modified proposal of (Itti et al., 1998). From the robot's camera, images are distributed across a number of computers and a set of filters is applied to the original stream at each node in the first line of processors. Each of them corresponds to one type of retinal feature maps, which are calculated at different scales. Within each feature processor, maps at different scales are combined to generate a global conspicuity map that emphasizes locations that stand out from their surroundings. The conspicuity maps are combined into a global saliency map, which encodes the saliency of image locations over the entire feature set. The time-integrated global saliency map is supplied as an input to a winner-take-all neural network, which is used to compute the most salient area in the image stream.

2.1 Generation of saliency maps

We have implemented the following feature processors on our system: color, intensity, orientation, motion, and disparity (see also Figure 2). Especially the generation of disparity, motion, and orientation feature maps are time consuming processes and it would be impossible to implement and visualize all of them on one computer and in real-time. The most computationally expensive among them is the generation of orientation feature maps, which are calculated by Gabor filters. They are given by

$$\Phi(\mathbf{x}) = \frac{\|\mathbf{k}_{\mu,\nu}\|^2}{\sigma^2} \exp\left(-\frac{\|\mathbf{k}_{\mu,\nu}\|^2 \|\mathbf{x}\|^2}{2\sigma^2}\right)\left(\exp\left(i\mathbf{k}_{\mu,\nu}^{T}\mathbf{x}\right) - \exp\left(-\frac{\sigma^2}{2}\right)\right), \quad (1)$$

where $\mathbf{k}_{\mu,\nu} = k_\nu[\cos(\phi_\mu), \sin(\phi_\mu)]^T$. Gabor kernels were suggested to model the receptive fields of simple cells in primary visual cortex. In (Itti et al., 1998) a single scale k_ν and four orientations ϕ_μ = 0, 45, 90, 135, were used. It has been shown, however, that there exist simple cells sensitive not only to specific positions and orientations, but also to specific scales. We therefore applied Gabor kernels not only at four different orientations but also at four different scales. For the calculation of motion, we used a variant of Lucas-Kanade algorithm. A correlation-based technique was used to generate disparity maps at the available frame rate (30 Hz).

At full resolution (320 x 240), the above feature processors generate 2 feature maps for color (based on double color opponents theory (Sekuler and Blake, 2002)), 1 for intensity, 16 for orientation, 1 for motion and 1 for disparity. Center-surround differences were suggested as a computational tool to detect local spatial discontinuities in feature maps that stand out from their surround. Center-surround differences can be computed by first creating Gaussian pyramids out of the initial feature maps. From the uppermost scale $\mathbf{I}_f(0)$, where f is the corresponding feature, maps at lower scales are calculated by filtering of the map at the previous scale with a Gaussian filter. The resolution of a map at lower scale is half the resolution of the map at the scale above it. Center-surround differences are calculated by subtracting pyramids at coarser scale from the pyramids at finer scale. For this calculation

the pyramid maps at coarser scales are up-sampled to finer scales. We calculated the center-surround differences between the pyramids $\mathbf{I}_f(c)$, c = 2, 3, 4, and $\mathbf{I}_f(s)$, $s = c + \Delta$, Δ = 2, 3. This results in 6 maps per feature.

The combination of center-surround differences into conspicuity maps for color $\mathbf{J}_c(t)$, intensity $\mathbf{J}_b(t)$, orientation $\mathbf{J}_o(t)$, motion $\mathbf{J}_m(t)$, and disparities $\mathbf{J}_d(t)$ at time t involves normalization to a fixed range and searching for global and local maxima to promote feature maps with strong global maxima. For each modality, center-surround differences are combined into conspicuity maps at the coarsest scale. This process is equivalent to what has been implemented by (Itti et al., 1998) and we omit the details here. The conspicuity maps are finally combined into a global saliency map $\mathbf{S}(t)$

$$\mathbf{S}(t) = w_c\mathbf{J}_c(t) + w_b\mathbf{J}_b(t) + w_o\mathbf{J}_o(t) + w_m\mathbf{J}_m(t) + w_d\mathbf{J}_d(t)\,. \tag{2}$$

The weights w_c, w_b, w_o, w_m, and w_d can be set based on top-down information about the importance of each modality. In the absence of top-down information, they can be set to a fixed value, e. g. 0.2, if all five features are to have the same influence. Finally, to deal with a continuous stream of images, the saliency maps need to be time-integrated

$$\mathbf{S}_{int}(t) = \gamma^{\delta}\mathbf{S}_{int}(t-\delta) + \mathbf{G}_{\sigma} * \mathbf{S}(t),\ 0 < \gamma < 1, \tag{3}$$

where $\delta \geq 1$ is the difference in the frame index from the previous saliency map and $\mathbf{G}_{\sigma} * \mathbf{S}(t)$ is the convolution of the current saliency map with the Gaussian filter with standard deviation σ.

2.2 Winner-take-all network

The aim of the preattentive processing is to compute the currently most salient area in the image so that the robot's eye can saccade towards this area and place it into the center of the fovea, thus enabling the robot to dedicate its computational resources to the processing of the foveal image area in the next processing step. Winner-take-all network has been suggested as means to calculate the focus of attention from the saliency map (Koch and Ullmann, 1987). We use the leaky integrate-and-fire model to build a two layer 2-D neural network of first order integrators to integrate the contents of the saliency map and choose a focus of attention over time. It is based on the integration of the following system of differential equations:

$$\begin{aligned} \frac{du_1}{dt}(\mathbf{x},t) + \frac{1}{\tau_1}u_1(\mathbf{x},t) &= \sum_{\mathbf{y}} w_1(\mathbf{x},\mathbf{y})\mathbf{S}_{int}(\mathbf{y},t) - \sum_{\mathbf{y}} w_{co}(\mathbf{x},\mathbf{y})u_2(\mathbf{y},t), \\ \frac{du_2}{dt}(\mathbf{x},t) + \frac{1}{\tau_2}u_2(\mathbf{x},t) &= \sum_{\mathbf{y}} w_2(\mathbf{x},\mathbf{y})u_1(\mathbf{y},t) \end{aligned} \tag{4}$$

where $u_i(\mathbf{x}, t)$ is the membrane potential of the neuron of the i-th layer located at $\mathbf{x}$ at time t, τ_i is the time constant of the i-th layer, w_i is the weighting function of the lateral connections of the i-th layer between locations $\mathbf{x}$ and $\mathbf{y}$ and $w_{co}(\mathbf{x},\mathbf{y})$ is the weighting function of connections between the first and the second layer. Functions $w_i(\mathbf{x},\mathbf{y})$ are given by:

$$w_1(\mathbf{x},\mathbf{y}) = w_2(\mathbf{x},\mathbf{y}) = \frac{1}{2\pi\sigma_{\text{inh}}^2}\exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma_{\text{inh}}^2}\right). \quad (5)$$

Function $w_{\text{co}}(\mathbf{x},\mathbf{y})$ models the coupling effects between the neurons of the network including long-range inhibition and short-range excitation to produce the winning neuron. It is defined as:

$$w_{\text{co}}(\mathbf{x},\mathbf{y}) = \frac{c_{\text{in}}^2}{2\pi\sigma_{\text{in}}^2}\exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma_{\text{in}}^2}\right) - \frac{c_{\text{ex}}^2}{2\pi\sigma_{\text{ex}}^2}\exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma_{\text{ex}}^2}\right) \quad (6)$$

We used Euler's method to integrate Equations (4). The integration frequency was set to 100 Hz and is higher than the timing of the vision signal (30 Hz). Hence before updating the integrated saliency map $\mathbf{S}_{\text{int}}$, Equations (4) are integrated a few times as temporal smoothing. When the potential of one of the neurons of the second layer $u_2(\mathbf{x}, t)$ reaches the adaptive firing threshold, the robot eyes move so that the most salient area is placed over the fovea. Vision processing is suppressed during the saccade. Since postattentive processing has not been integrated into the system yet, the robot just waits for 500 ms before moving its eyes back to the original position. At this point the neurons of the second layer are reset to their ground membrane voltage as global lateral inhibition and a local inhibitory signal is smoothly propagated from the first to the second layer at the attended location as inhibition of return. The strength of the inhibitory effect is gradually reduced as the time passes to allow for further exploration of the previously attended regions.

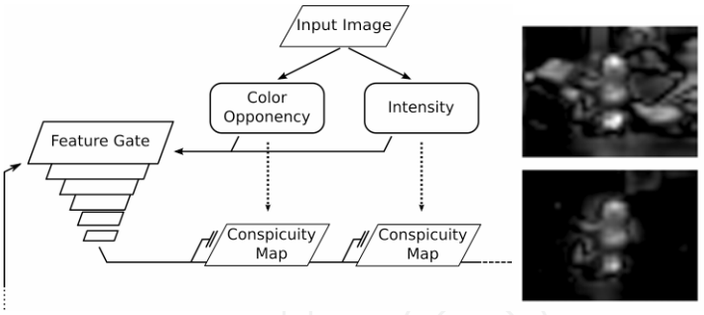


Figure 3. The FeatureGate top-down biasing system added to the simplified architecture from Figure 2. The top-down feature vectors are fed to the FeatureGate system, which finds the inhibitory signals for the conspicuity maps (created in parallel by the bottom-up system). To the right, the resulting saliency map (upper right) and color opponency conspicuity map with inhibition from the FeatureGate subsystem

3. Top-down guidance

As already mentioned in Section 2, feature-wide top-down effects can be introduced into the system by selecting different weights when combining the conspicuity maps into a single saliency map by means of Eq. (2). A recent model by (Navalpakkam and Itti, 2006) computes

optimal weights based on the observer's prior beliefs about the scene (target and distractors) to arrive at the linear combination of feature maps that best separates the sought for feature from its expected background. Boosting certain types of features over the others is, however, still a broad mechanism, best suited for biasing higher-level search towards certain kind of data, and not well suited for pinpointing specific features.

Another approach is to introduce context-dependent spatial restrictions on the search, with inhibition on areas not likely to have features the system searches for. (Balkenius et al., 2004) present an adaptive contextual system where the conspicuity map content at the current fixation serves as the contextual cue as to where, in absolute or relative terms, the desired feature is likely or unlikely to be. The saliency map is boosted or inhibited accordingly. This kind of mechanism is more specific, in that it can explicitly focus on, or disregard, areas independently of its bottom-up saliency.

If the goal is to introduce top-down influences looking for specific features, we need a different kind of mechanism. More precisely, we want to be able to give a particular feature vector and bias the saliency towards points that in some way or another match that feature vector. One way to accomplish this has been proposed in FeatureGate model of human visual attention (Cave, 1999). This model introduces top-down effects by lateral inhibition of activation in feature maps. At every given point, the inhibition is a function of this point's nearness to the expected feature vector as compared to the nearness of neighboring points to the same feature vector. The measure of nearness must be defined by a suitable metrics ρ. A point receives inhibition when a neighboring area is closer to the target top-down feature *tf* than the current location **x**. The model conversely boosts points proportionally to their distinctiveness at each level (defined as the sum of absolute differences to the neighboring points). Top-down inhibition and local distinctiveness are weighted and combined. The results are gated up from fine to coarse scales, effectively increasing the spatial extent of the inhibition within each level, finally resulting in a pyramid of inhibitory values for different spatial scales.

Let $N_c(\mathbf{x})$ be the neighborhood of location **x** at level *c* in the pyramid and let $S_c(\mathbf{x})$ be all pixels in the neighborhood that are closer to the target than **x**:

$$S_c(\mathbf{x}) = \left\{ \mathbf{y} \in N_c(\mathbf{x}) ; \rho\left(\mathbf{I}_{tf}(\mathbf{y};c)\right) < \rho\left(\mathbf{I}_{tf}(\mathbf{x};c)\right) \right\}, \tag{7}$$

Let $\mathbf{I}_{tf}(0)$ be the map generated by processing the image with the top-down target feature processor at full resolution. The top-down inhibition $\mathbf{I}_{tf}^{d}$ is calculated as the value proportional to the difference in the distance from the target feature

$$\mathbf{I}_{tf}^{d}(\mathbf{x};c) = \sum_{\mathbf{y} \in S_c(\mathbf{x})} \left| \rho\left(\mathbf{I}_{tf}(\mathbf{y};c)\right) - \rho\left(\mathbf{I}_{tf}(\mathbf{x};c)\right) \right|, \tag{8}$$

For each *j*, *j* = c, m, i, o, d, we (optionally) also calculate the distinctiveness $\mathbf{I}_{j}^{d}$

$$\mathbf{I}_{j}^{d}(\mathbf{x};c) = \sum_{\mathbf{y} \in S_c(\mathbf{x})} \left| \mathbf{I}_{j}(\mathbf{y};c) - \mathbf{I}_{j}(\mathbf{x};c) \right|. \tag{9}$$

We obtain the signal for inhibition by weighting these two measures

$$\mathbf{I}_j^{\text{inh}}(\mathbf{x};c) = \alpha \mathbf{I}_j^d(\mathbf{x};c) - \beta \mathbf{I}_{tf}^d(\mathbf{x};c). \tag{10}$$

The next, coarser pyramid level is constructed by comparing each point in a small neighborhood $N_{c-1}(\mathbf{x}')$ (2x2 points by default) at the previous level and propagating only the least inhibited point to the point $\mathbf{x}$ at the next level:

$$\mathbf{I}_{tf}(\mathbf{x};c) = \mathbf{I}_{tf}(\mathbf{y};c-1),\ \mathbf{I}_j(\mathbf{x};c) = \mathbf{I}_j(\mathbf{y};c-1),\ \mathbf{y} = \underset{\mathbf{y}\in N_{c-1}(\mathbf{x}')}{\arg\max}\left\{\mathbf{I}_j^{\text{inh}}(\mathbf{y};c-1)\right\}. \tag{11}$$

The process is repeated until we get - at the top-level of the pyramid - a single element representing the globally most salient point with respect to the bottom-up map and, optionally, the distinctiveness. The level below contains the most salient point in each of the four quadrants of the image and so forth. The actual values do not encode how good the matches are as they are relative to other points in the image. With $\alpha = 0$ we get a pure top-down system well adapted for use together with a separate bottom-up system. However, the proposed computational mechanism for the integration of both systems described below is robust enough so that the precise settings are not very important for the overall functionality. It is also possible to set α and β in such a way that the system behaves similarly to the one described by (Cave, 1999) and these were the parameters used in our experiments.

To integrate the result of the above algorithm into the saliency map that can be supplied to the winner-take-all network, we generate a second conspicuity map based on the position (at the appropriate pyramid level) of the most salient top-down point $\mathbf{x}_j$ with respect to the feature map j and the given top-down feature vector. The following formula is used to generate this second map

$$\mathbf{M}_j(\mathbf{x};t) = \begin{cases} 1, & \mathbf{x} = \mathbf{x}_j \\ \mathbf{M}_j(\mathbf{x};t-1) * 0.9, & \mathbf{x} \neq \mathbf{x}_j \end{cases}. \tag{12}$$

Top-down influence	Total fixations	Target fixations	Target fixations percentage
0%	62	23	37%
10%	65	31	48%
20%	53	30	57%
30%	33	19	58%
40%	20	14	70%
50%	19	16	84%
75%	14	13	93%
100%	6	6	100%

Table 1. Fixation data

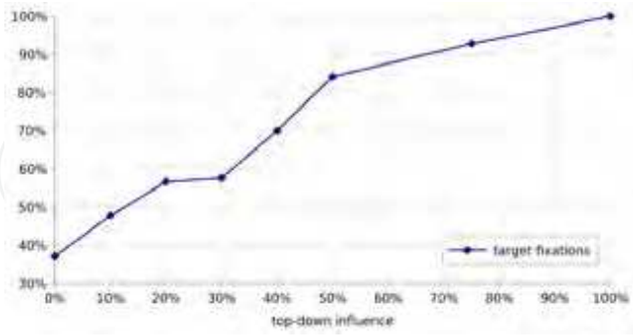


Figure 4. Target fixations as a function of top-down influence. A 30 second image sequence was run through the system with different influence settings. The attended object is fairly salient by itself with 37% of fixations when using the bottom-up saliency system only. The top-down system is able to rapidly boost this ratio, with almost 85% of all fixations when λ is at 0.5

Finally, a new conspicuity map is computed by adding the weighted top-down and bottom-up conspicuity maps $\mathbf{J}_j'(t) = \lambda\mathbf{M}_j(t) + (1-\lambda)\mathbf{J}_j(t)$. Thus the relative importance of bottom-up and top-down saliency processing is determined by the parameter λ. In Figure 3, $\lambda = 0.5$ was used and $\mathbf{M}_j$ were initially set to zero, i. e. $\mathbf{M}_j(0) = 0$.

We ran a series of tests to check the effects of top-down biasing. A short image sequence of about 30 seconds depicting an object (teddy bear) being moved around was used as input to the system. In these experiments the system used color opponency and intensity as low-level features and did not generate saccades. The shifts in current region of interest were recorded; note that the saccades that would be performed are selected from a subset of these covert attentional shifts. The top-down system was primed with a vector roughly matching the brightness and color space position of the target object. Given proper weighting factors, the locations selected by FeatureGate are close to the intended target with high probability. On the other hand, by keeping the bottom-up cue in the system we ensure that very salient areas will be attended even if they don't match the feature vector.

Tests were run with all settings equal except for the parameter λ specifying the influence of the top-down system relative to the bottom-up saliency. The data generated is presented in Table 1. We tested the system from 0% influence (only the bottom-up system active) to 100% (only the top-down system used). Fewer saccades are generated overall if there exists a dominant target in the image matching the feature vector and the influence of the top-down cue is high. Since in such cases the behavior of the system changes little as we increase the top-down influences, we tested the system only at two high top-down settings (75% and 100%). Figure 4 demonstrates that the system works much as expected. The target object is fairly salient but it is fixated on less than 40% of the time if only bottom-up saliency is used. With top-down biasing the proportion of fixations spent on the target increases rapidly and with equal influence the target is already fixated 84% of the time. At high levels of top-down influence the target becomes almost totally dominant and the object is fixated 100% of the time when $\lambda = 1$. The rapid dominance of the target as we increase the top-down influence is natural as it is a salient object already. Note that if the top-down selection mechanism has several areas to select from - as it will if there are several objects matching the top-down criteria or if the object has a significant spatial extent in the image - the effect of the top-

down system will spread out and weaken somewhat. Also, with two or more similar objects the system will generate saccades that occasionally alternate between them as the inhibition of return makes the current object temporarily less salient overall.

The above experiment was performed with a top-down system closely following the original FeatureGate model in design. Specifically, we still use the distinctiveness estimate at each level. Alternatively, we could apply only the top-down inhibitory mechanism and simply use the map $\mathbf{I}_{tf}^{d}(\mathbf{x};c)$ of Eq. (8) - calculated at the same pyramid level c as the conspicuity maps $\mathbf{J}_j(t)$ - to generate the inhibitory signal. In many practical cases, the behavior of such a system would be very similar to the approach described above, therefore we do not present separate experiments here.

65911	65910	65907	65911	65912	61250	61249	61250	61250	61250	70656	70656	70656	70656	70656
65912	65910	65910	65911	65913	61251	61251	61250	61251	61251	70675	70675	70675	70675	70675
65912	65912	65910	65912	65914	61252	61251	61250	61251	61252	70678	70678	70678	70678	70678
65913	65912	65910	65913	65915	**61253**	**61253**	**61250**	**61253**	**61253**	70695	70695	70695	70695	70695
65914	**65912**	**65910**	**65913**	**65916**	61253	61253	61254	61254	61254	70711	70711	70711	70711	70711
65915	65914	65913	65915	65917	61255	61253	61254	61254	61255	70715	70715	70715	70715	70715
65917	65914	65913	65916	65918	61256	61256	61254	61256	61256	**70724**	**70724**	**70724**	**70724**	**70724**
65918	**65916**	**65913**	**65916**	**65919**	61257	61256	61257	61257	61257	**70757**	**70757**	**70757**	**70757**	**70757**
65918	65916	65916	65918	65920	61258	61258	61257	61257	61258	70758	70758	70758	70758	70758
65919	65918	65916	65919	65921	61259	61258	61257	61259	61259	70777	70777	70777	70777	70777
65920	**65918**	**65916**	**65921**	**65922**	61260	61260	61260	61260	61260	70790	70790	70790	70790	70790
65921	65921	65919	65922	65923	61260	61260	61260	61261	61261	70799	70799	70799	70799	70799
65923	65921	65919	65922	65924	61262	61262	61260	61261	61262	70802	70802	70802	70802	70802
65924	**65923**	**65919**	**65923**	**65925**	**61263**	**61262**	**61260**	**61263**	**61263**	70815	70815	70815	70815	70815
65925	65923	65922	65923	65926	61264	61264	61264	61264	61264	70837	70837	70837	70837	70837

Table 2.. Frame indices of simultaneously processed images under different synchronization schemes. In each box, ordered from left to right clumn, the frame indices belong to the disparity, color, orientation, intensity, and motion conspicuity map. See text in Section 4.1 for further explanations

4. Synchronization of processing streams

The distributed processing architecture presented in Figure 2 is essential to achieve real-time operation of the complete visual attention system. In our current implementation, all of the computers are connected to a single switch via a gigabit Ethernet. We use UDP protocol for data transfer. Data that needs to be transferred from the image capture PC includes the rectified color images captured by the left camera, which are broadcast from the frame grabber to all other computers on the network, and the disparity maps, which are sent directly to the PC that takes care of the disparity map processing. Full resolution (320 x 240 to avoid interlacing effects) was used when transferring and processing these images. The five feature processors send the resulting conspicuity maps to the PC that deals with the calculation of the saliency maps, followed by the integration with the winner-take-all network. Finally, the position of the most salient area in the image stream is sent to the PC taking care of motor control. The current setup with all the computers connected to a single gigabit switch proved to be sufficient to transfer the data at full resolutions and frame rates. However, our implementation of the data transfer routines allows us to split the network

into a number of separate networks should the data load become too large. This is essential if the system is to scale to a more advanced vision processing such as shape analysis and object recognition.

A heterogeneous cluster in which every computer solves a different problem necessarily results in visual streams progressing through the system at different frame rates and with different latencies. In the following we describe how to ensure smooth operation under such conditions.

4.1 Synchronization

The processor that needs to solve the most difficult synchronization task is the one that integrates the conspicuity maps into a single saliency map. It receives input from five different feature processors. The slowest among them is the orientation processor that could roughly take care of only every third frame. Conversely, the disparity processor works at full frame rate and with lower latency. While it is possible to further distribute the processing load of the orientation processor, we did not follow this approach because our computational resources are not unlimited. We were more interested in designing a general synchronization scheme that allows us to realize real-time processing under such conditions.

The simplest approach to synchronization is to ignore the different frame rates and latencies and to process the data that was last received from each of the feature processors. Some of the resulting frame indices for conspicuity maps that are in this case combined into a single saliency map are shown in the leftmost box of Table 2. Looking at the boldfaced rows of this column, it becomes clear that under this synchronization scheme, the time difference (frame index) between simultaneously processed conspicuity maps is quite large, up to 6 frames (or 200 milliseconds for visual streams at 30 Hz). It does not happen at all that conspicuity maps with the same frame index would be processed simultaneously.

Ideally, we would always process only data captured at the same moment in time. This, however, proves to be impractical when integrating five conspicuity maps. To achieve full synchronization, we associated a buffer with each of the incoming data streams. The integrating process received the requested conspicuity maps only if data from all five streams was simultaneously available. The results are shown in the rightmost box of Table 2. Note that lots of data is lost when using this synchronization scheme (for example 23 frames between the two boldfaced rows) because images from all five processing streams are only rarely simultaneously available.

We have therefore implemented a scheme that represents a compromise between the two approaches. Instead of full synchronization, we monitor the buffer and simultaneously process the data that is as close together in time as possible. This is accomplished by waiting that for each processing stream, there is data available with the time stamp before (or at) the requested time as well as data with the time stamp after the requested time. In this way we can optimally match the available data. The algorithm is given in Figure 5. For this synchronization scheme, the frame indices of simultaneously processed data are shown in the middle box of Table 2. It is evident that all of the available data is processed and that frames would be skipped only if the integrating process is slower than the incoming data streams. The time difference between the simultaneously processed data is cut to half (maximum 3 frames or 100 milliseconds for the boldfaced rows). However, the delayed synchronization scheme does not come for free; since we need to wait that at least two

frames from each of the data streams are available, the latency of the system is increased by the latency of the slowest stream. Nevertheless, the delayed synchronization scheme is the method of choice on our humanoid robot.

Request for data with frame index n:
  get access to buffers and lock writing
  $r = 0$
  for $i = 1,\ldots,m$
    find the smallest $b_{i,j}$ so that $n < b_{i,j}$
    if such $b_{i,j}$ does not exist
      reply images with frame index n not yet available
      unlock buffers and exit
    if $b_{i,(j-1)\%M} \leq n$
      $j_i = b_{i,(j-1)\%M}$
    else
      $r = \max(r, b_{i,j})$
  if $r > 0$
    reply r is the smallest currently available frame index
    unlock buffers and exit
  return $\{ \mathbf{P}_{1,j_1}, \ldots, \mathbf{P}_{m,j_m} \}$
  unlock buffers and exit

Figure 5. Pseudo-code for the delayed synchronization algorithm. m denotes the number of incoming data streams, or - in other words - the number of preceding nodes in the network of visual processes. To enable synchronization of data streams coming with variable latencies and frame rates, each data packet (image, disparity map, conspicuity map, joint angle configuration, etc.) is written in the buffer associated with the data stream, which has space for M latest packets. $b_{i,j}$ denotes the frame index of the j-th data packet in the buffer of the i-th processing stream. $P_{i,j}$ are the data packets in the buffers and m is the number of data streams coming from previous processes

We note here that one should be careful when selecting the proper synchronization scheme. For example, nothing less than full synchronization is acceptable if the task is to generate disparity maps from a stereo image pair with the goal of processing scenes that change in time. On the other hand, buffering is not desirable when the processor receives only one stream as input; it would have no effect if the processor is fast enough to process the data at full frame rate, but it would introduce an unnecessary latency in the system if the processor is too slow to interpret the data at full frame rate. The proper synchronization scheme should thus be carefully selected by the designer of the system.

5. Robot eye movements

Directing the spotlight of attention towards interesting areas involves saccadic eye movements. The purpose of saccades is to move the eyes as quickly as possible so that the spotlight of attention will be centered on the fovea. As such they constitute a way to select task-relevant information. It is sufficient to use the eye degrees of freedom for this purpose. Our system is calibrated and we can easily calculate the pan and tilt angle for each eye that are necessary to direct the gaze towards the desired location. Human saccadic eye movements are very fast. The current version of our eye control system therefore simply moves the robot eyes towards the desired configuration as fast as possible.
Note that saccades can be made not only towards visual targets, but also towards auditory or tactile stimuli. We currently work on the introduction of auditory signals into the proposed visual attention system. While it is clear that auditory signals can be used to localize some events in the scene, the degree of cross-modal interactions between auditory and visual stimuli remains an important research issue.

6. Conclusions

The goals of our work were twofold. On the one hand, we studied how to introduce top-down effects into a bottom-up visual attention system. We have extended the classic system proposed by (Itti et al., 1998) with top-down inhibitory signals to drive attention towards the areas with the expected features while still considering other salient areas in the scene in a bottom-up manner. Our experimental results show that the system can select areas of interest using various features and that the selected areas are quite plausible and most of the time contain potential objects of interest. On the other hand, we studied distributed computer architectures, which are necessary to achieve real-time operation of complex processes such as visual attention. Although some of the previous works mention that parallel implementations would be useful and indeed parallel processing was used in at least one of them (Breazeal and Scasselatti, 1999), this is the first study that focuses on issues arising from such a distributed implementation. We developed a computer architecture that allows for proper distribution of visual processes involved in visual attention. We studied various synchronization schemes that enable the integration of different processes in order to compute the final result. The designed architecture can easily scale to accommodate more complex visual processes and we view it as a step towards a more brain-like processing of visual information on humanoid robots.
Our future work will center on the use of visual attention to guide higher-level cognitive tasks. While the possibilities here are practically limitless, we intend to study especially how to guide the focus of attention when learning about various object affordances, such as for example the relationships between the objects and actions that can be applied to objects in different situations.

7. Acknowledgment

Aleš Ude was supported by the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) funded by the European Commission.

8. References

Balkenius, C., Åström, K. & Eriksson, A. P. (2004). Learning in visual attention. *ICPR 2004 Workshop on learning for adaptable visual systems*, Cambridge, UK.

Breazeal, C. & Scasselatti, B. (1999). A context-dependent attention system for a social robot. *Proc. Sixteenth Int. Joint Conf. Artificial Intelligence*, Stockholm, Sweden, pp. 1146-1151.

Cave, K. R. (1999). The FeatureGate model of visual selection. *Psychological Research*, 62:182–194.

Driscoll, J. A.; Peters II, R. A. & Cave, K. R. (1998). A visual attention network for a humanoid robot. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Victoria, Canada, pp. 1968–1974.

Itti, L.; Koch, C. & Niebur E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(11) :1254–1259.

Koch C. & Ullman S. (1987). Shifts in selective visual attention: towards the underlying neural circuitry. *Matters of Intelligence*, L. M. Vaina, Ed., Dordrecht: D. Reidel Co., pp. 115–141.

Navalpakkam, V. & Itti, L. (2006). An integrated model of top-down and bottom-up attention for optimizing detection speed. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, New York, pp. 2049-2056.

Rolls, E. T. & Deco, G. (2003). *Computational Neuroscience of Vision*. Oxford, University Press.

Sekuler, R. & Blake, R. (2002). *Perception*, 4th ed. McGraw-Hill.

Stasse, O.; Kuniyoshi Y. & Cheng G. (2000). Development of a biologically inspired real-time visual attention system. *Biologically Motivated Computer Vision: First IEEE International Workshop*, S.-W. Lee, H. H. Bülthoff, and T. Poggio, Eds., Seoul, Korea, pp. 150–159.

Sun, Y. & Fisher, R. (2003). Object-based visual attention for computer vision. *Artificial Intelligence,* 146(1):77-123.

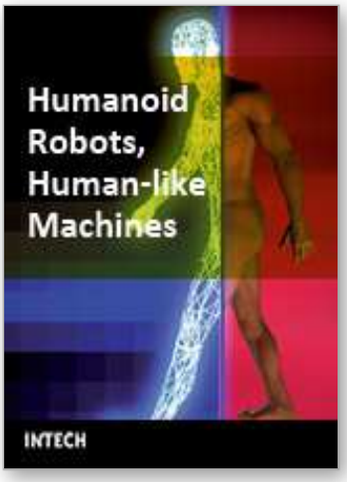
Treisman, A. M. & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12(1) :97–136.

Tsotsos, J. K. (2005). The selective tuning model for visual attention. *Neurobiology of Attention*. Academic Press, pp. 562–569.

Vijayakumar, S.; Conradt, J.; Shibata, T. & Schaal, S. (2001). Overt visual attention for a humanoid robot. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Maui, Hawaii, USA, pp. 2332–2337.

Wolfe, J. M. (2003). Moving towards solutions to some enduring controversies in visual search. *Trends in Cognitive Sciences*, 7(2):70–76.

Yarbus, A. L. (1967) Eye movements during perception of complex objects. In: *Eye Movements and Vision*, Riggs, L. A. (Ed.), pp. 171–196, Plenum Press, New York.



Humanoid Robots, Human-like Machines

Edited by Matthias Hackel

ISBN 978-3-902613-07-3

Hard cover, 642 pages

Publisher I-Tech Education and Publishing

Published online 01, June, 2007

Published in print edition June, 2007

In this book the variety of humanoid robotic research can be obtained. This book is divided in four parts: Hardware Development: Components and Systems, Biped Motion: Walking, Running and Self-orientation, Sensing the Environment: Acquisition, Data Processing and Control and Mind Organisation: Learning and Interaction. The first part of the book deals with remarkable hardware developments, whereby complete humanoid robotic systems are as well described as partial solutions. In the second part diverse results around the biped motion of humanoid robots are presented. The autonomous, efficient and adaptive two-legged walking is one of the main challenge in humanoid robotics. The two-legged walking will enable humanoid robots to enter our environment without rearrangement. Developments in the field of visual sensors, data acquisition, processing and control are to be observed in third part of the book. In the fourth part some "mind building" and communication technologies are presented.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen