Modulating Vision with Motor Plans: A Biologically-inspired Efficient Allocation of Visual Resources

Luka Lukic, Aude Billard and José Santos-Victor

Abstract—**This paper presents a novel, biologically-inspired, approach for an efficient management of computational resources for visual processing. In particular, we modulate a visual "attentional landscape" with the motor plans of a robot. The attentional landscape is a more recent, general and a more complex concept of an arrangement of spatial attention than a simple ''attentional spotlight'' or a ''zoom-lens'' model of attention. A higher attention priority for visual processing must be given to manipulation-relevant parts of the visual field, in contrast with other, manipulation-irrelevant, parts. Hence, in our model visual attention is not exclusively defined in terms of visual saliency in color, texture or intensity cues, it is rather modulated by motor (manipulation) programs. This computational model is supported by recent experimental findings in visual neuroscience and physiology. We show how this approach can be used to efficiently distribute limited computational resources devoted to visual processing, which is very often the computational bottleneck in a robot system. The model offers a view on the well-know concept of visual saliency that has not been tackled so far, thus this approach can offer interesting alternative prospects not only for robotics, but also for computer vision, physiology and neuroscience. The proposed model is validated in a series of experiments conducted with the iCub robot, both using the simulator and with the real robot.**

I. INTRODUCTION

Vision is one of the most computationally demanding modules in a robot system, representing very often a bottleneck for manipulation applications. Most of the approaches in robot vision are based on standard image processing techniques, ignoring most, if not all, the task-relevant dynamic information. This implies that the visual system and the arm-hand system are usually considered as two independent modules that communicate only in the direction from vision to manipulation, which implies that during visual processing the valuable information from the manipulation system is completely ignored. In this work we show that coupling visual processing with manipulation plans can drastically improve visual performances, in particular, the speed of visual computation.

If we put this in a real-world context, let us imagine a robot bartender, equipped with an active stereo camera system that has the task to grasp a glass, fill it with a

Luka Lukic and José Santos-Victor are with Computer and Robot Vision Laboratory (VISLAB), IST, Portugal: `luka.lukic@epfl`, `jasv@isr.ist.utl.pt`.

Aude Billard is with Learning Algorithms and Systems Laboratory (LASA), EPFL, Switzerland: `aude.billard@epfl`.

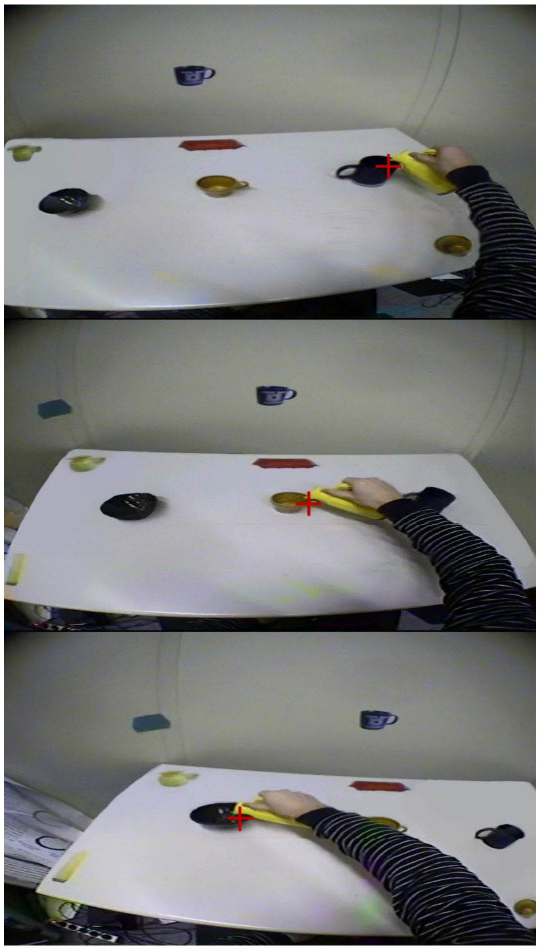

Figure 1. Experimental setup with a natural task. The subject is instructed to pour the tea into two cups and one bowl that are placed close to the horizontal midline of the table. 4 pictures of various objects are placed close to the border of the table and 2 pictures are placed on the wall facing the subject. These pictures play the role of visually salient distractors because they share the same visual features with the objects, but remain completely irrelevant for manipulation through the entire task. The *overt attention*, i.e. gaze movements, together with the scene as viewed from the subject's standpoint are recorded by using the WearCam system [1]. The order of the figures from top to bottom corresponds to the progress of the task. The cross superposed on the video corresponds to an estimated gaze position. It can be seen that the gaze is tightly bound to an object that is relevant to spatio-temporal requirements of the task. In spite of the presence of salient distractors, the gaze remains tightly locked on the current object of interest. This behavior cannot be predicted by the feature-based saliency maps, even with the top-down extensions because in manipulation tasks, perceptual processing is biased towards manipulation-relevant regions of the visual field, not towards the most textured or distinctively colored stimulus.

beverage of choice, and serve it to a guest. In a visually-aided manipulation, based on standard vision processing approach, during reaching and grasping for the target object, in every cycle of the control loop vision scans every part of both stereo images searching for the target object and potential obstacles, in order to update the robot's knowledge about their state (position, orientation and other properties of interest that might change during a task). Assume that the motion of the arm has been initiated and is directed toward a specific object, say a wine glass (the obstacles will by definition be all objects that obstruct an intended movement). Here, a question arises: why would one want to scan the peripheral parts of the stereo images for obstacles, since these correspond to regions in the workspace ten meters or so from the wine glass that is at around 30 cm from the hand? Clearly the space scanned should be restricted to region of the space that are task-relevant.

Contrary to robots, humans are able to rapidly and graciously perform complicated tasks with a limited amount of computational resources. One of the reasons for the human performance is an efficient distribution of the visual resources to select only relevant information for reaching and grasping among the plethora of visual information. Humans are able to efficiently and routinely manage this challenging task of selective information processing, in a seemingly effortless manner, by means of highly customized attentional mechanisms. In visual attention, two mechanisms are recognized: *covert attention* and *overt attention* [2]. Covert visual attention corresponds to an allocation of mental resources for processing extrafoveal visual stimuli. Overt visual attention consists in active visual exploration involving saccadic eye movements (Fig. 1). These two mechanisms are instantiations of the same underlying mechanism of visual attention, hence intermingled both functionally and structurally, working in synchronization and complementing each other. Covert attention selects interesting regions in the visual field, which are subsequently attended with overt gaze movements for high-acuity foveated extraction of information [3]. Furthermore, visual attention (covert and overt) is tightly coupled with manipulation. Numerous findings from visual neuroscience and physiology provide evidence that visual attention is bound and actively tailored with respect to spatio-temporal requirements of manipulation tasks [4]–[8]. Fig. 1 illustrates how attention is drawn towards manipulation-relevant regions of the visual field, even in a common, well-known natural task such as tea serving.

In this paper, we hypothesize that such a biologically-inspired, explicit, active adaptation of attention with respect to motor plans can endow robot vision with a mechanism for efficient allocation of limited visual resources. This approach contributes to the state of the art in visual-based reaching and grasping, tackling visual attention from a new, alternative perspective where visual saliency is not defined in terms of low-level visual features such as color, texture or intensity of the visual stimuli, but rather in terms of manipulation-relevant parts of the visual field as salient regions. In our model, the attentional mechanism becomes a fundamental building element of the motor planning system. At each cycle of the control loop, the visual and motor systems modulate each other sending each other control signals. The proposed approach is evaluated in robotic experiments using the iCub humanoid robot [9].

The rest of the paper is structured as follows. Section II reviews related work on computational modeling of visual attention, its use in robotics, and surveys the biological evidence onto which we ground our approach to tackle the existing problems. Section III describes our computational model and system architecture. Section IV reports on validations of the approach in experiments with the iCub robot. In Section V we conclude this work.

II. RELATED WORK

A. Computational modeling of attention

Most of the current work on computational modeling of attention is related to the feature integration theory of attention from physiology [10]. The feature integration theory advocates the idea that low-level, pre-attentive features capture attention. The intuition behind this approach is that a non-uniform spatial distribution of features is somehow correlated with their informative significance. The influence of the low-level features on capturing attention is motivated by functions of the neural circuitry in the early primate vision and experimental findings in scene observation tasks [11]. The most influential computational implementation grounded on this theory is the concept of the saliency map [12]. Low-level features such as color, orientation, brightness and motion are extracted in parallel from the visual input. In computational models, the visual input is represented as a digitized 2D image. Low-level features from the visual stimuli compete across image space and multiple spatial scales building spatial banks of features that correspond to center-surround contrast computed across different scales. The feature banks are normalized and combined by a weighted sum to create the master saliency map. The focus of attention is driven by the interplay between a winner-take-all mechanism (WTA) and an inhibition of return mechanism (IOR) that operates on the final saliency map. This pure bottom-up, stimulus driven approach has been subsequently extended to guided visual search by an additional weighting of the feature channels with a top-down bias that comes from the prior knowledge about objects [13].

B. Attention in robotics

Related work in robotics is strongly influenced by the aforementioned computational models of attention. Whereas most of the computational models assume covert attention shifts, i.e. no movements of the head and the eyes are involved, most robots are equipped with an active camera system, which makes them suitable for active, overt visual exploration. Robotic applications inherently rely on a saliency map-based scheme to evaluate visual stimuli, and then, instead of shifting covert focus of attention, they actively initiate saccadic movements of the cameras to bring the fixation to the most salient point in the visual field [14]. A number of robotic applications are primarily concerned with implementing saliency maps in order to achieve biologically-inspired saccadic and smooth-pursuit eye movements either with a single pan-tilt camera or a complete robot head [15]. These schemes have been extended to biologically inspired log-polar vision [16]. Saliency-based attention has been investigated in conjunction with exploration, development and learning for humanoid robots [16]. Attentional-based vision has been addressed as an aid to sociable robots to improve human-robot interaction [17] and in imitation learning [18].

C. Current shortcomings of attention-based models for robot vision

Although the efforts made in the robotic community have been very fruitful, expanding theoretical foundations and providing practical applications of attentional mechanisms, the most prominent use of attentional schemes still remains confined to object tracking, scene exploration, mimicking the human visual system for robotic studies of development and for providing human-like visual behavior for sociable robots. The use of attention for active, real-time vision-based manipulation that relies on reliable visual information at each cycle of the control loop is still very limited. This is an issue we aim to address in this work. In particular we identify the following three issues as critical: i) speed of computation, ii) focus of attention and iii) salient features.

1) Speed of computation: Attention in primates evolved as a mechanism to select a small subset of low-level visual information for further, high-level processing. This is done in order to efficiently allocate limited computational resources. However, as previously mentioned, most work in robotics related to attention is inspired by the saliency model of Itti and Koch [12]. Regardless of the massively parallel architecture, constructing a saliency map is a computationally heavy task. The best reported times on CPU-based implementations are of an order of ~50 ms for a single map [19], the time which doubles for a stereo system, after which some high-level visual processing is done in the later stages in the visual processing pipeline. This limits the applicability of the classical saliency map approach for fast real-world robotic problems such as real-time adaptation to perturbations in grasping tasks with obstacle avoidance.

2) Focus of attention: The majority of models of attention assume that a focus of attention, the so-called attentional spotlight, is a circular shaped region of a fixed radius [20], which is centered at a point with the highest saliency in the visual field. Zoom-lens models extend the attentional spotlight concept by allowing the radius of an attentional "window" to change with respect to task demands [21]. Both the spotlight and zoom-lens models restrict applicability of attentional mechanisms for real-world robotic scenarios in complex tasks because only one location in the visual field is (covertly) selected as the focus of attention, towards which the further attentional interest is oriented (covertly or overtly). A number of recent studies from visual neuroscience and physiology suggest that covert attention can take on a complex spatial arrangement [22]. Baldauf et al. have found that covert attention supports pre-planning of a sequence of movements towards multiple reaching goals, by distributing peaks of attention along an intended reaching path [5], [6]. These findings show that covert attention can be distributed not only at one location, as overt attention, but rather simultaneously form a complex "attentional landscape" in the visual field.

3) Salient features: Computational models of attention have shown good performances and significant statistical similarity to human strategies in simple scene viewing and in guided search tasks [11], [12], but describing human gaze behavior in more complex tasks is far beyond their capabilities. We hypothesize that this is due to the fact that only low-level image features are taken into account by the models that compute salience, whereas a strong top-down bias from motor planning is completely ignored. This is rather surprising, considering that there are numerous evidences that report on very strong coupling between motor planning and attention allocation. In studies that used overt gaze movements as a measure of attention, researchers have found that the gaze is driven by spatio-temporal task demands in simple navigation tasks [23], by manipulation in natural, well-known tasks [4], in moderately complex tasks involving obstacle avoidance [24], and in very complex tasks such as ball sports [25]. Similarly, studies that analyzed distribution of covert attention have shown similar results. Covert attention is drawn to objects relevant to manipulation, even when reaching for multiple targets in a sequence [5], or in parallel by engaging bimanual manipulation [6]. All these studies suggest that low-level feature-based saliency is suppressed when humans are engaged in visually-aided physical tasks, regardless whether the task is manipulation or navigation, whether the interaction with the object is performed in a parallel or in a sequential manner, and regardless whether gaze movements are suppressed or not. In simple words, in physical tasks manipulation-relevant parts of the visual field are visually salient.

We proceed further with the explanation of the architecture of our computational model. We develop the model by drawing inspiration from the aforementioned physiological studies that report on strong coupling between visual attention and motor planning. By equalizing motion-relevant as attention-salient we aim to tackle the reviewed current weaknesses in the existing attention models.

III. COMPUTATIONAL APPROACH AND SYSTEM ARCHITECTURE

We first describe our robotic eye-arm-hand controller and motor planning mechanism. A learned eye-arm-hand Coupled Dynamical System (CDS) is used in order to "mentally simulate" the consequences of intended actions, more specifically, to compute (i.e. plan) an intended trajectory and to identify obstacles. We then explain how we transform the mentally-simulated trajectory to the image planes of stereo cameras. The projected mentally-simulated trajectory is used to compute an attentional landscape, i.e. a saliency map. Finally, we describe how this covert attentional landscape is used for top-down object search, how it is related to gaze movements and control of reaching and grasping by using the CDS framework.

A. *Eye-arm-hand controller*

We here briefly review the structure of an eye-arm-hand controller, developed in our previous work [26], which we use to generate robot motion and to plan the arm-hand reaching trajectory.

The controller is based on the CDS framework, where the main idea is to estimate separate Autonomous Dynamical Systems (DS) that correspond to each body part (one DS for the eyes, one for the arm and one for the hand), and then couple them explicitly. This controller consists of five building "blocks": three dynamical systems and two coupling blocks between them. They are organized in the following order: eye dynamics $\rightarrow$ eye-arm coupling $\rightarrow$ arm dynamics $\rightarrow$ arm-hand coupling $\rightarrow$ hand dynamics, where the arrow direction indicates the direction of control signals. The gaze DS is the master to the arm DS, and the arm DS is the master to the hand DS. Fig. 2 illustrates the architecture of the CDS. We used the human motion capture data recorded in a reaching-and-grasping task to estimate the parameters of the controller [26]. The time-invariant properties of the CDS allow rapid adaptation to spatial and temporal perturbations, where the explicit coupling between each dynamical system ensures that their behavior is correctly synchronized, even when the motion is abruptly perturbed far from the motion recorded in human demonstrations.

This framework allows us to perform visuomotor coordination in the presence of an obstacle, as well. We use the CDS mechanism in order to mentally simulate the consequences of planned arm movements, specifically to detect objects that obstruct the intended reach-for-grasp actions and to identify them as obstacles. An obstacle is used as the intermediary target for the visuomotor system, which allows us decompose the obstacle avoidance task in two segments: from the start to the obstacle and from the obstacle to the target. During obstacle avoidance, the primary modulation of the arm is controlled in Cartesian space, which, together with controlled hand preshape, ensures that the end-effector avoids the obstacle. The desired arm joint postures suitable for obstacle avoidance are provided to the IK solver. Treating the obstacle as an intermediary target results in a simple and computationally lightweight scheme for obstacle avoidance. We use the mentally-simulated trajectory in order to bias visual resources to manipulation-relevant parts of the visual field, which we describe in the next section.

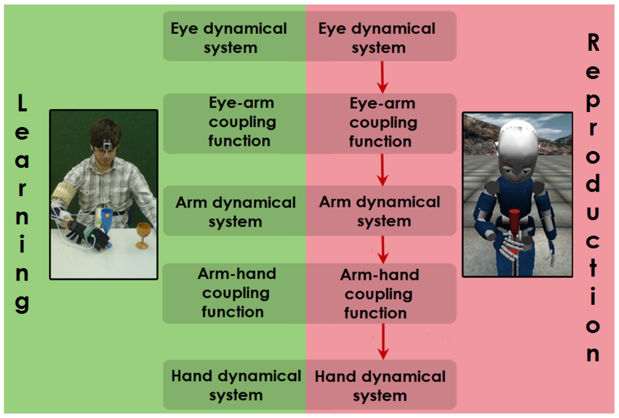

Figure 2. CDS-based eye-arm-hand controller. Left (green) part of the figure shows how the CDS model is learned. Reproduction of motion on the robot is shown on the right side of the figure (red part). CDS consists of five building "blocks": three dynamical systems (the eyes, the arm and the hand) and two coupling models: eye-arm coupling and arm-hand coupling.

B. *Workspace to the image plane projection*

In order to be able to distribute visual attention with respect to manipulation plans, we need to obtain a transformation that will map an arbitrary point from the workspace to the image plane. In particular, we want to map a set of points of a mentally-simulated trajectory of the arm to the image planes of both cameras of the robot.

1) Approach: The mentally-simulated trajectory of the arm, from the current position towards the final position at the current time t, is represented as $x_t^n \in \mathbb{R}^3, \forall n \in [1, N_t]$, where N_t represents the total number of discrete samples. This mentally-simulated trajectory at every cycle of the control loop t is obtained from the CDS controller, explained in Section III-A. The kinematic configuration at the current time t of the torso-neck-arm is represented with the torso, neck, and head joints $q_t \in \mathbb{R}^9$, $\forall t$. The transformation function is of the form:

$$p_{t,i}^n = f_i(c_t^n), \; \forall n \in [1, N_t] \,, \qquad (1)$$

where $c_t^n \in \mathbb{R}^{12}$, $c_t^n = \begin{bmatrix} x_t^n \\ q_t \end{bmatrix}$, and $p_{t,i}^n \in \mathbb{R}^2$ represents the projection of the trajectory to the image plane of the i-th camera, where $i = \{left, right\}$.

A naive, straightforward approach would be to compute a sequence of kinematic transformations through the torso-neck-head kinematic chain in order to obtain the extrinsic camera parameters, and use them together with the intrinsic parameters of the camera to obtain this projection. The problem associated with this approach is that the kinematic parameters of the real robot do not exactly correspond to the mathematical model of the robot kinematic. Even when these differences are small, they accumulate when propagated through the kinematic chain.

Another alternative is to rely on a machine learning approach. The robot would explore in a babbling-like manner a set of random kinematic configurations, and during this exploration it would segment an object (e.g. a small colored ball) set at a randomly chosen position from a set of known positions in the workspace. The data obtained during the exploration (encoder readings of the joints in the torso-neck-head chain, the position of the object in the workspace and its projection to the camera planes) would be saved and used to learn a mapping function. A problem associated with this approach is that the babbling-like exploration with the real robot is very costly because in order to build a reliable estimate of this nonlinear mapping, the size of a training set needs to be arbitrarily large to be representative, usually of an order a few thousands data samples.

We take here an intermediary step that is a compromise between the two previously described approaches. The idea is to use the simulator of a robot in order to obtain a large number of training samples by employing babbling, and use this data set to estimate initial set parameters of the mapping model. Because we know that the differences between the simulated model and the real robot are not too big, this model is then incrementally adapted with the data obtained from the real robot, which accounts for only a small fraction of the data obtained in the simulator.

2) Neural network approach to the workspace-camera transformations: A feed-forward neural network is a suitable machine learning algorithm for our application for several reasons [27]. Feed-forward neural networks can compute multi-input-multi-output functions. Their output is fast to compute in real-time because the computation consists of a short sequence of matrix-vector multiplications, followed by (non)linear transfer functions. Feed-forward neural networks are suitable for incremental learning, either in the batch or in the stochastic, online mode. This allows us to first estimate this function from the data in the simulator, and then adapt it with the data from the real robot.

The parameters of an architecture of neural networks for transformation from the workspace to the image coordinates (i.e. number of layers and the number of hidden units, etc) are determined by using grid-search on the mean squared error (MSE) between the recorded image projections and retrieved projections from the model. We tested 10 different architectures, and for every architecture we performed 10 learning runs in order to achieve robustness with respect to random initialization of network parameters. We used the Levenberg–Marquardt optimization algorithm with early-stopping in order to prevent overfitting [27]. The recorded data set is randomly partitioned for 70 % of the data devoted to training, 15 % data for validation, and 15 % data for testing. The lowest MSE on the testing set is achieved with two hidden layers with 25 nodes in each hidden layer. Transfer functions in the hidden layer are hyperbolic tangent sigmoid, and in the output layer are linear. The data set is normalized to have zero mean and unity variance. In order to get the real-time performances, a network class is implemented in C++ by using linear algebra functions from OpenCV library [28]. The time needed to transform 30 trajectory points by using neural nets to the image planes of both cameras is $\sim$1 ms.

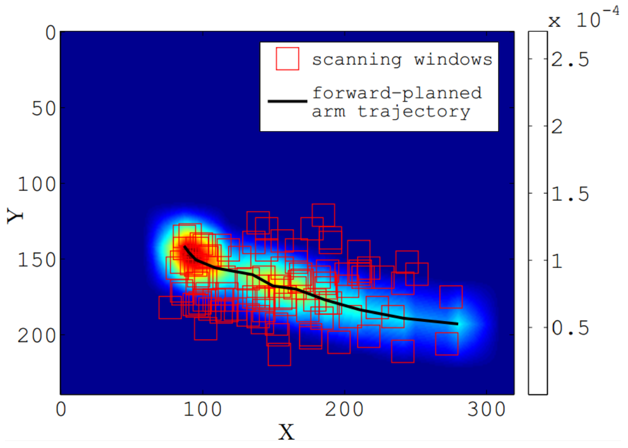

Figure 3. Nonuniform image processing based on a manipulation-relevance saliency. The black line corresponds to a mentally-simulated (forward-planned) arm trajectory that is projected to the image plane of a stereo camera. The heat-map corresponds to an intensity of the saliency function. The red rectangles are scanning image windows, for which visual features are computed. The anchors of the scanning windows are sampled with respect to a saliency function, i.e. more dense visual scanning is done where the relevance function has higher values, and less dense scanning where it has low values. The plot corresponds to the right stereo camera. The same procedure is done for the left camera.

C. Attentional landscape

After we project the mentally-simulated trajectory to the image planes, we construct an attentional landscape which associates high saliency close to the mentally-simulated trajectory perceived in the image coordinates. To compute the attentional landscape, i.e. a measure of visual processing priority (saliency map), we use a bivariate kernel smoothing function [29], where kernels are placed at every point of the projection of the mentally-simulated trajectory to the image planes. Formally, we compute a saliency map for each camera i as follows:

$$I_i(p) = \frac{1}{N_t h^h h^v} \sum_{n=1}^{N_t} K(p - p_{t,i}^n), \tag{2}$$

where $p \in \mathbb{R}^2$ corresponds to two-dimensional pixel coordinates of the image plane, $K(p - p_{t,i}^n) = k\left(\frac{p^h - p_{t,i}^{n,h}}{h^h}\right) k\left(\frac{p^v - p_{t,i}^{n,v}}{h^v}\right)$, $k(.)$ represents a kernel and h^h and h^v are kernel widths along the horizontal and vertical image dimensions. We use triangular kernels because they are faster to compute than Gaussian kernels. The triangular kernel is expressed as follows: $k(z) = \begin{cases} 1 - \left|z\right|, & \left|z\right| \leq 1 \\ 0 & otherwise \end{cases}$. Kernel smoothing function assigns high values of saliency close to the mentally-simulated trajectory projected to the images planes of stereo cameras, which decreases in the directions away from the trajectory (Fig. 3). The attentional landscape is used to guide image processing in order to efficiently distribute limited visual resources. The part of the image with higher saliency draws more visual processing, and the opposite is true. In the next section we explain how we distribute visual processing with respect to saliency.

D. Image processing

We propose here two ways to use the attentional landscape to guide visual processing. Our approach is general enough to be used as a pre-modulating technique to any kind of standard image processing detectors and segmentation techniques (pixel-by-pixel color segmentation, Viola-Jones, SIFT, SURF, etc).

1) Thresholding: One simple approach suitable for pixel-by-pixel color processing and interest point detectors-descriptor approaches is to distribute visual processing to the region of the image where the saliency map $I_i(p)$ is higher or equal that some threshold d_i. It is easy to empirically estimate the computational time for processing the entire image and from this estimate cost per pixel. By sorting pixels with respect to ascending values of their saliency, we can pick a number of pixels corresponding to the available computational resources. From this sorted array, we can easily read the threshold d_i on the saliency. An approximate value of the threshold can be determined in ~3 ms for 4800 subsampled pixels by using the Quick Sort algorithm.

2) Sampling: A number of image processing techniques employ image processing within a scanning window, e.g. Viola-Jones detector, histogram-based detector, Rowley-Baluja-Kanade detector, etc. Here the task is to determine position of the scanning windows with respect to a saliency map, in order to have more dense scanning where the saliency is large, and less dense scanning in spatial regions with low saliency. Because we use a kernel smoothing function to build a saliency map, we can treat the saliency map as a bivariate probability density function and use any kind of sampling techniques to sample spatial locations of scanning windows. Again, we can empirically obtain a cost associated to process the image in each window, and from the total visual resources calculate determine the number of points to sample from the saliency map. We use the inverse transform sampling method [29] and perform this independently for the vertical and horizontal axes of the image. The procedure is simple and works as follows:

1) compute a cumulative sum along each axis independently: $S_i(a) = \Sigma_{j=1}^{a} I_i(p^a)$, where $a = \{v,\, h\}$
2) draw a random number $\tilde{u}_a$, $a = \{v,\, h\}$ from the uniform distribution in the interval [0, 1]
3) compute axis coordinates as: $\tilde{a} = S_i^{-1}(\tilde{u_a})$
4) go to step 2) and repeat for a given number of scanning windows.

The inverse transform sampling based on implementation with look-up tables as C-arrays can be computed and sampled from in ~1 ms for an image 320×240 for 50 sampled scanning windows (Fig. 3). It is noteworthy that we recompute the saliency maps at every cycle. This implies that there is no need to implement the IOR mechanism and deal with the problems of the change of coordinates associated with standard saliency models [14], which simplifies our approach and hence reduces the overall computational time.

E. From covert attentional landscape to overt eye movements and manipulation

As explained in the previous section, when the saliency map is constructed, the top-down visual scan is performed in the spatial regions that have high salience. These two stages correspond to covert visual attention. After the targets (and/or obstacles) are detected, the overt gaze movements are initiated towards the first intermediary target in a synchronous manner together with the arm and the hand motion by using our CDS eye-arm-hand controller [26]. In a no-obstacle task, the eye-arm-hand system is directly driven towards the target. In tasks with obstacle avoidance the eye-arm-hand system is driven towards the obstacle, which is treated as an intermediary target for the visuomotor system, as explained in Section III-A. When the obstacle is avoided, the system is driven towards the object to be grasped.

IV. VALIDATION OF THE APPROACH

We validate our method in the iCub simulator and with the real robot with a task of reaching for and grasping a kitchenware object. Resolution of the cameras in the stereo setup is 320×240. We test this approach with two standard image processing techniques. For the first visual detector, we select a scanning window hue-saturation histogram-based detector. We implement this detector by using functions from OpenCV library [28]. The computation times are averaged over 100 measurements. The cumulative time for calculating a projection of the mentally-simulated trajectory to the image plane, computing a saliency map and sampling from it for both cameras is on average 24.5 ms. We choose to only process image with 50 image windows of the size 20×20. Visual processing for 50 sampled 20×20 windows takes on average 28 ms. These times sum up to the total time of 52.5 ms for our method. The classical method with an uniformly sliding window, for the same setup, on average requires 174 ms, therefore our method reduces computational time by a factor of ~3.1 ×, saving 121.5 ms in every loop cycle and still successfully completing the task (Fig. 4). The times are reported in Table I.

For the second detector, we selected SURF [30]. SURF is a powerful detector because it provides visual features that are robust to moderate changes of the perspective. Because it computes feature point descriptors, it allows to detect partially occluded objects. However, SURF (together with a family of similar detectors like SIFT, GLOH, etc.) is very computationally demanding, with the cost being double for a binocular system, hence it has limited applicability for manipulation where the stereo vision is used in the loop. In our runs, it takes on average 526 ms to compute SURF

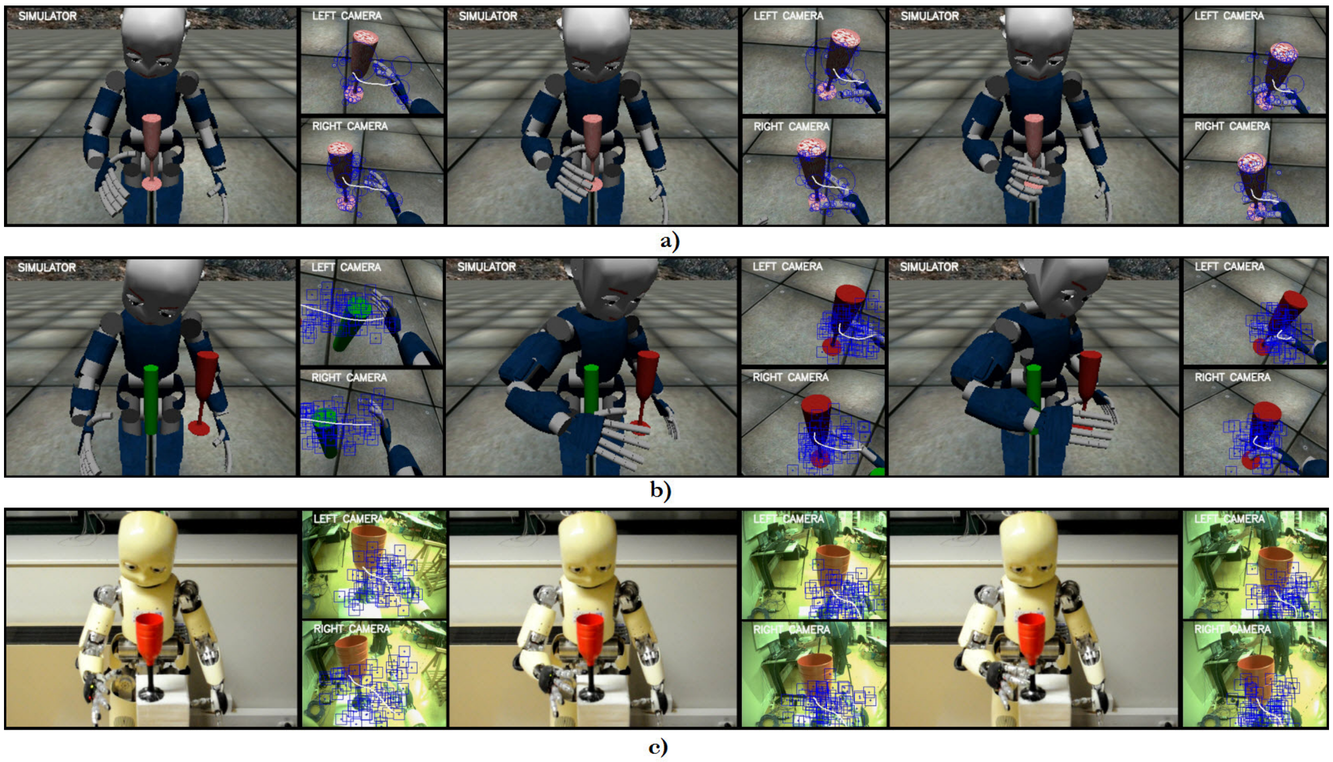

Figure 4. Experiments of visually-guided reaching and grasping in the iCub's simulator, in two different scenarios with two detectors, and with the real robot. Visual computation is in every cycle of the of the control loop distributed to manipulation-relevant spatial locations of the stereo images. Figures show execution of eye-arm-hand coordination from the start of the task (left) until the successful grasp completion (right). The white line corresponds to a mentally-simulated arm trajectory that is projected to the image planes of stereo cameras. The top row of figures (a) corresponds to the no-obstacle scenario with SURF detector. The blue circles correspond to detected strong feature points. The middle row (b) corresponds to the obstacle scenario with histogram-based detector. The blue rectangles are scanning image windows, for which visual features are computed. The bottom row (c) corresponds to the no-obstacle scenario with histogram-based detector with the real robot. Figures show execution of eye-arm-hand coordination from the start of the task (left) until the successful grasp completion (right).

for the full stereo image pair[1] To test our method, we chose to devote only 30 % of the total visual resources to processing, i.e. to process only 30 % of the image pixels with the highest salience. The cumulative time for calculating a projection of the mentally-simulated trajectory to the image plane, computing the saliency map and thresholding it for both cameras is on average 30 ms. Visual processing for the region of images with the highest salience takes on average 269.5 ms. These times sum up to the total time of 299.5 ms, saving 226.5 ms in every loop cycle compared to computing SURF for the whole image. The time of 226.5 ms is still large for some real-time applications, but if we choose to intelligently process the images, adjusting image resources to manipulation plans, we can speed up computation by a factor of ~1.8 $\times$, as we show it in Table II. Fig. 4 shows the experiments with the simulator and the real robot.

Finally, it is important to mention that, in addition to speeding up visual processing, this approach facilitates the accuracy of visual detections during an ongoing prehensile movement. The common problem with visual detections in cluttered scenes (as the one in Fig. 4c) is that there is a significant number of false positives. Because we bound visual processing to motor plans of a robot, we significantly reduce false positive detections. In the context of the conducted experiments, there are no false positive detections of the objects (the target and the obstacle) in the parts of visual field that are irrelevant to motor plans, because the relevant objects are not likely to be there.

The experiments presented here are available in the video submitted with this paper.

V. CONCLUSIONS

In this paper we presented an approach where a saliency is computed with respect to current manipulation plans. This approach is inspired from recent works from physiology and visual neuroscience that suggest that attention is bound to spatio-temporal manipulation requirements and that covert attention forms the so-called attentional landscape, which is a more complex and a more general concept than an attentional spotlight. We show that this approach can help to efficiently distribute limited visual resources in a robot system, significantly reducing resources compared to classical uniform image processing, but still allowing for a robot to perform complicated tasks, such as manipulation with perturbations

[1]We used the implementation available from OpenCV library.

projecting trajectory with neural nets	saliency map	sampling	sparse image processing	**total time with our approach**	**standard (full) image processing approach**	**savings with our approach**
1	22.5	1	28	**52.5**	**174**	**121.5 (3.1 x)**

Table I
CPU TIME FOR HISTOGRAM SLIDING-WINDOW DETECTOR

projecting trajectory with neural nets	saliency map	thresholding	sparse image processing	**total time with our approach**	**standard (full) image processing approach**	**savings with our approach**
1	22.5	6.5	269.5	**299.5**	**526**	**226.5 (1.8 x)**

Table II
CPU TIME FOR SURF

CPU time required to compute motion-relevant saliency map and perform sparse histogram-based image processing (Table I) and SURF computation (Table II) image with our approach compared to standard-uniform processing with the same detectors. We are able to save 121.5 ms for the histogram-based detector and 226.5 ms for SURF in every cycle compared for to the standard approach. The time in the tables is provided in milliseconds, and corresponds to computation for both cameras of the binocular setup of the iCub robot with 320×240 color input images. The times are averaged over 100 frames.

and obstacle avoidance.

VI. ACKNOWLEDGMENTS

This work was supported in part by the FCT doctoral grant (SFRH/BD/51072/2010).

REFERENCES

[1] B. Noris, J. Keller, and A. Billard, "A wearable gaze tracking system for children in unconstrained environments," *Computer Vision and Image Understanding*, 2010.

[2] J. S. Werner and L. M. Chalupa, *The visual neurosciences*. Bradford Book, 2004.

[3] S. Liversedge and J. Findlay, "Saccadic eye movements and cognition," *Trends in Cognitive Sciences*, vol. 4, no. 1, pp. 6–14, 2000.

[4] M. Hayhoe, A. Shrivastava, R. Mruczek, and J. Pelz, "Visual memory and motor planning in a natural task," *Journal of Vision*, vol. 3, no. 1, 2003.

[5] D. Baldauf, M. Wolf, and H. Deubel, "Deployment of visual attention before sequences of goal-directed hand movements," *Vision Research*, vol. 46, no. 26, pp. 4355–4374, 2006.

[6] D. Baldauf and H. Deubel, "Visual attention during the preparation of bimanual movements," *Vision Research*, vol. 48, no. 4, pp. 549–563, 2008.

[7] W. S. Geisler, "Visual perception and the statistical properties of natural scenes," *Annu. Rev. Psychol.*, vol. 59, pp. 167–192, 2008.

[8] D. Baldauf and H. Deubel, "Attentional selection of multiple goal positions before rapid hand movement sequences: An event-related potential study," *Journal of Cognitive Neuroscience*, vol. 21, no. 1, pp. 18–29, 2009.

[9] Metta *et al.*, "The icub humanoid robot: An open-systems platform for research in cognitive development," *Neural Networks*, vol. 23, no. 8-9, pp. 1125–1134, 2010.

[10] A. M. Treisman and G. Gelade, "A feature-integration theory of attention," *Cognitive Psychology*, vol. 12, no. 1, pp. 97–136, 1980.

[11] P. Reinagel and A. M. Zador, "Natural scene statistics at the centre of gaze," *Network: Computation in Neural Systems*, vol. 10, no. 4, pp. 341–350, 1999.

[12] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[13] V. Navalpakkam and L. Itti, "Modeling the influence of task on attention," *Vision Research*, vol. 45, no. 2, pp. 205–231, 2005.

[14] M. Begum and F. Karray, "Visual attention for robotic cognition: a survey," *IEEE Transactions on Autonomous Mental Development*, vol. 3, no. 1, pp. 92–105, 2011.

[15] L. Manfredi, E. S. Maini, P. Dario, C. Laschi, B. Girard, N. Tabareau, and A. Berthoz, "Implementation of a neurophysiological model of saccadic eye movements on an anthropomorphic robotic head," in *IEEE-RAS International Conference on Humanoid Robots*, 2006.

[16] F. Orabona, G. Metta, and G. Sandini, "Object-based visual attention: a model for a behaving robot," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*. IEEE, 2005.

[17] C. Breazeal, A. Edsinger, P. Fitzpatrick, and B. Scassellati, "Active vision for sociable robots," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 31, no. 5, pp. 443–453, 2001.

[18] M. Ogino, H. Toichi, Y. Yoshikawa, and M. Asada, "Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 414–418, 2006.

[19] S. Kestur, M. S. Park, J. Sabarad, D. Dantara, V. Narayanan, Y. Chen, and D. Khosla, "Emulating mammalian vision on reconfigurable hardware," in *IEEE 20th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2012.

[20] M. I. Posner, C. R. Snyder, B. J. Davidson *et al.*, "Attention and the detection of signals," *Journal of Experimental Psychology*, vol. 109, no. 2, pp. 160–174, 1980.

[21] C. W. Eriksen and J. D. S. James, "Visual attention within and around the field of focal attention: A zoom lens model," *Perception & Psychophysics*, vol. 40, no. 4, pp. 225–240, 1986.

[22] D. Baldauf and H. Deubel, "Attentional landscapes in reaching and grasping," *Vision Research*, vol. 50, no. 11, pp. 999–1013, 2010.

[23] C. Rothkopf and D. Ballard, "Image statistics at the point of gaze during human navigation," *Visual Neuroscience*, vol. 26, no. 01, pp. 81–92, 2009.

[24] R. Johansson, G. Westling, A. Bäckström, and J. Flanagan, "Eye–hand coordination in object manipulation," *The Journal of Neuroscience*, vol. 21, no. 17, pp. 6917–6932, 2001.

[25] M. F. Land and P. McLeod, "From eye movements to actions: how batsmen hit the ball," *Nature Neuroscience*, vol. 3, no. 12, pp. 1340–1345, 2000.

[26] L. Lukic, J. Santos-Victor, and A. Billard, "Learning coupled dynamical systems from human demonstration for robotic eye-arm-hand coordination," IEEE-RAS International Conference on Humanoid Robots (Humanoids), Osaka, Japan, 2012.

[27] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*. Prentice Hall, 1998.

[28] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Incorporated, 2008.

[29] K. P. Murphy, *Machine Learning: a Probabilistic Perspective*, 2012.

[30] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Lecture Notes in Computer Science*, vol. 3951, pp. 404–417, 2006.